Project Number. GFP0608

RUNNER'S LOG AND PREDICTIVE PERFORMANCE ANALYTICS

A Major Qualifying Project Report:

submitted to the faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by:

Alexander L. White

Date: June 1, 2007

Approved:

Professor Gary F. Pollice, Major Advisor

1. running log
2. performance prediction
3. data-centric model

Abstract

This report, prepared for the Worcester Polytechnic Institute, describes the development of a running log application and the development and analysis of a data-centric approach to running performance prediction. The java application incorporated common UI principles as well as a community aspect to facilitate and encourage its use. The data-centric predictive model was developed by parsing meet results to follow each individual’s performances. Simplified, predictions are created by analyzing individuals who have performed similarly to the input. As tested with 1148 male track performances and 1265 female track performances, the data-centric approach provided predictions with an average error of 3.05 percent for men and 3.63 percent for women. These errors are approximately 9 percent and 20 percent lower, respectively, than the leading “Purdy Points” model.

Acknowledgements

Gary Pollice

Patrick Hoffman

Pete Riegel

Dave Cameron

Jack Daniels and J. R. Gilbert

J. Gerry Purdy and J. B. Gardner

Run-Down.com: *Performance Predictors*

DirectAthletics.com – for their excellent database of meet results

Eric Yee of RunningAHEAD.com – for his web-based running log development

WPI Cross Country and Track Teams

Table of Contents

List of Illustrations

List of Tables

1. Introduction

In sports, those who compete or are fans share a great passion for analysis. It is in human nature to perform comparisons, pondering questions such as: “Is LeBron James better than Michael Jordan?” Unfortunately, these questions are often subjective and rely on a large number of factors. Running is more unique in that these factors are greatly restricted. The sport of track and field is about individual performance at its root and the single most important factor is the event, often associated with distance. In each event, athletes compete to see who can run faster or throw further or jump higher – a single measure decides the best. Since events are standardized, one need not compete directly against another to determine who performs the best for a given event. What about performances in different events? Often the only difference is the length that is run, making quantitative comparisons possible.

The field of running performance has become an obsession amongst many runners and analysts. It is trivial to determine who is better for a single distance as time will suffice. Comparing different distances becomes much more interesting. After the Atlanta Olympics in 1994, there was a debate between Michael Johnson and his 200 meter dash and Donovan Bailey and his 100 meter dash. While each could run the other’s event, it may not be that individual’s best distance. The ultimate hope is to provide evidence as to who performed the best. From an individual’s perspective, one could use these models as a basis to determine the distance at which he or she performs the best.

Perhaps more interesting is the application of these comparisons for predictive purposes. For example, if a man runs a mile in 5 minutes, what will his time be for two miles? He could run the two mile event however track seasons are generally short, sometimes only five meets. This makes it difficult and sometimes wasteful to try a range of races especially as a coach may need that individual to score points in specific events. By using these predictive models, one could predict instead for an approximation. These predictions are also useful for pacing if one were to run that distance. While a handful of performance and predictive models currently exist, each has its own strengths and weaknesses. For example most models fail to differentiate between male and female runners, some cater to elite performances, and others are intended for a certain type of distance. I hypothesize that female runners' performances span a greater range than those of males and thusly are not as well predicted by these models. I also hypothesize that the relative performance, elite versus average, of individuals affects the models' predictive behavior. I propose a data-centric methodology that utilizes existing runners' performances to predict another's.

The addition of a running log to this project was intended to, through its use by runners, provide data that could give insight into important factors for running performance. I felt that existing running logs were inadequate for this purpose. They are generally cumbersome to use or did not provide features that would encourage use and this is ultimately prohibitive for data analysis as potentially useful data would not be recorded. The recorded data, through mining techniques, could then be used to enhance the accuracy of the predictive model.

Even today, new training methods have been devised as human beings are very complex and the best method may not have been found. Additionally, it is often not a "one-size-fits-all" plan for runners, who can take years of experimentation to determine what training style provides better results. The data gathered from a running log may possibly validate different training methods, such as high mileage, fartlek runs, and other strategies.

The following Background section provides information that is important to the motivation for and development of this project, namely the state of existing running logs and performance models. The Methodology section documents the process I used to develop my predictive models. It also includes the approach and implementation details of the running log application under development. The Results and Analysis section presents the results of validation of existing models as well as my own. A brief discussion is provided for the running log as an early prototype in ongoing development. The last section or Future Work and Conclusions summarizes the results of the predictive models and recommends actions for further model research and analysis and improvements for the current state of the running log.

2. Background

This section describes information that is pertinent to the scope of my running log and the understanding of the performance models. A discussion of existing performance and predictive models is presented along with their strengths and shortcomings. The concept of running logs is introduced. Summaries of existing popular running logs are provided along with user feedback. Currently, there is a need for a running log that incorporates the best features and improves upon existing designs. Additionally, the onset of the Internet age has greatly simplified a data-centric method for creating a predictive model and, of equal importance, a method for validating predictive models.

A distinction has been made between performance models and predictive models. A performance model is designed to relate the quality of comparable efforts across different events. For example, is a man who runs a 100 meter dash in 10 seconds “better” than one who runs a 4 minute mile? These models can be used for prediction as most runners do not stray far out of their area of events: sprints, mid-distance, or distance. A predictive model is designed to give a specific runner an idea of what performance to expect for new distances based on previous performances.

2.1. Performance Models

The following performance models have arisen out of running analysis and have been published in a variety of media from email correspondences to magazines. Beyond being used for comparison or prediction as previously described, another use may be to determine “equivalent” qualifying standards for post-season competitions. This may help ensure equally sized running fields as an event that has an easier qualifying time would unnecessarily balloon the field. Some of the models below are not used in comparison to my model due to integration difficulty or scope and are stated as such. They are included for completeness and as a starting point for future research and analysis. I would like to acknowledge Run-Down.com for their useful compilation of performance models and predictive models to be discussed in the next subsection. Their web calculator and brief explanations can be found at this URL: http://run-down.com/statistics/calc.php.[1]

2.1.1. David F. Cameron's Model

Also known as Dave Cameron's Model, he developed this model which was published via email correspondence with other performance analysts. He began by compiling "a handful" of top performances from the U.S. and world levels over a range of distance from 400 meters to 50 miles. Using non-linear regression, he fit seven acceptable models to the data. Validating these models with older data, he somewhat subjectively picked one that performed the most accurately.[2] This model, like most, performs on an input of a single performance, a distance and time pair, and a desired output distance. A speed versus distance basis is used as speed behaves more linearly with changes in distance than time, allowing the speed to be multiplied by a computer factor. The formula on this model is as follows where `old_dist` is the distance run, `old_time` is the time run, and `new_dist` is the new distance the performance would like to be compared to:

[1] Performance Predictors, 2007, 16 May 2007 <http://run-down.com/statistics/calc.php>.

[2] Time-equivalence Model: David F. Cameron Model, Jun 1998, 16 May 2007 <http://www.cs.uml.edu/~phoffman/cammod.html>.

```
a = 13.49681 - (0.000030363 * old_dist) +
     (835.7114 / old_dist$^{0.7905}$)
b = 13.49681 - (0.000030363 * new_dist) +
     (835.7114 / new_dist$^{0.7905}$)
multiplier = a / b
new_time = (old_time / old_dist) * multiplier *
     new_dist
```

Figure 1: Pseudocode for Dave Cameron's model
[Source: Run-Down.com *Explaining the Performance Predictors*]

This model is obviously limiting in that no distances below 400 meters were used in fitting the model, so while predictions for these distance are allowed, they are not guaranteed to be appropriate. This becomes especially noticeable for the 100 meter dash which is heavily influenced by the runner's maximum velocity and the startup time to achieve that velocity. In fact the current world record for the 200 meter dash held by Michael Johnson (19.32 seconds) is *less* than twice the world record time for the 100 meter dash (9.77 seconds) shared by Asafa Powell and Justin Gatlin.[3] An additional concern is that because the model was fit with only elite men's data, it may not properly compare or predict female or average runners – a common concern.

2.1.2. Purdy Points Model

One of the oldest performance models, Purdy Points may be one of the most well-known performance models as it has been observed to be applicable for performance comparisons across all commonly run distances. Developed by J. Gerry Purdy and J. B. Gardner, the model was published in *Medicine and Science in Sports* in 1970 under the

[3] World Records - Men, 2 Dec. 2006, 16 May 2007 <http://www.trackandfieldnews.com/tfn/records/records.jsp?sex=M&typeId=0&listId=1>.

title “Computer generated track scoring tables.” The model relied on older running performances known as the “Portuguese Scoring Tables” compiled in 1936. The table lists the speeds for world record performances, up to 1936, for distances from 40 meters to 100,000 meters or roughly 62.14 miles. Each of these performances were recorded as speeds in a straight line once peak speed is reached and were deemed equal and given an arbitrary score of 950 points. The model operates on a distance and time pair and accounts for the startup time as well as the additional time needed to run around the curves of the track. The model determines world record speeds or standard speeds in meters per second from the table through linear interpolation.[4] The pseudocode for determining the slowdown from turns and startup is shown here with units in meters where `speed` is the interpolated speed.

```
frac = fraction of distance run on turns
// A 400m lap will have 200m of turns or 0.5
slowdown = 0.0065 * frac * speed * speed
// Turn slowdown is a function of speed squared
slowdown += 0.20 + 0.08 * speed
// A constant 0.2s is added with a smaller delay
// that varies with speed – it doesn’t take as
// long to reach top speed if it’s slower
```

Figure 2: Psuedocode for slowdown in Purdy Points Model
[Adapted from: Patrick Hoffman, *Gardner-Purdy points*]

The turns prove significant as this requires more effort due to the changes in direction and a general inability to run exactly on the inside of a lane. Doing some simple calculations shows that if one were to run in the middle of their lane around the turns, as

[4] J. B. Gardner and J. G. Purdy, "Computer generated track scoring tables," Medicine and science in sports 2.3 (1970): 152-61.

opposed to the inside, one would run an extra 3 meters per lap or 75 meters in a 10 kilometer race! Once this standard time with slowdown is calculated, scaling factors `A` and `B` are used to achieve the Purdy Points. These values were found by comparing speeds at distance of 100 meters and 3 miles for 950 point and 1035 point performances (the approximate Purdy Points for a 1970 world record).[5] It is important to see that these factors are not constant and adjust with the speed. The following pseudocode shows these final calculations where `Tp` is the input time run in seconds and `speed` is the interpolated speed used in the previous pseudocode.

```
Ts = Standard time from tables + slowdown
Tp = Performance time to be compared
k = 0.0654 - 0.00258 * speed
A = 85/k
B = 1 - 950/A       // 950 from point assignment
Purdy Points = A (Ts/Tp - B)
```

Figure 3: Psuedocode for points in Purdy Points Model
[Adapted from: Patrick Hoffman, *Gardner-Purdy points*]

To perform predictions or time comparisons once a point value is determined, a reverse lookup can be done with the point value, which is matter of simple algebra. The Java code for this Purdy Points Model as modified from Patrick Hoffman's C program can be seen in Appendix A.

Purdy published a second version of the Purdy Points in *Research Quarterly* in 1974 under the title of "Least squares model for the running curve." For this model, Purdy chose to utilize world record performances up to 1970 and create a running curve equation, for speed as a function of distance, as opposed to table lookups. This

[5] Gardner-Purdy points, 2004, 16 May 2007 <http://www.cs.uml.edu/~phoffman/xcinfo3.html>.

incorporates slowdowns due to startup and turns and greatly simplifies the model. The running curve equation is a sum of five exponential terms. The remainder of the point calculation remains the same as the psuedocode above with the exception of 1035 instead of 950 being used to determine the scaling factor B. The Java code for this latter Purdy Model as modified from Patrick Hoffman can be seen in Appendix B.

These two models are most appropriate to my research as they are devoted to track performances, but again still have a few concerns. The lack of female consideration in this model prevents female performances from being directly compared to male performances. For example, an elite female may score 900 points whereas an elite male may score close 1,100. Because this model is based on data from 1936 and the records have changed drastically since (many now score over 1,100 points), the comparisons to today's athletes may not be as suitable as better training methods may have produced unequal gains for different distances.

2.1.3. Performance Tables

The following set of performances tables are prevalent in international and elite competition but were not used in my analysis due to the time needed to translating the tables into software functions. Many of these tables are copyrighted and provided as copy-protected PDF files. Analysis of these tables would be a very interesting area of research with respect to more average athletes and those of varying ages.

To some extent, the beginning of performance models began with the inclusion of the Decathlon as an event, predominantly in the 1912 Olympics. To determine an overall winner, each event was scored and summed to a total. The scoring was done with a "scoring table" based upon a function that would attempt to weight each performance equally such that no one event would have more impact than another. In the first few Olympics, these functions were actually linear and based upon two points: the current records for each individual event and something akin to an average of junior performances. Beginning in 1920, the International Association of Athletics Federations (IAAF) began to examine the theory and merit behind the scoring tables, concluding that:

- "Each unit of improvement in an athlete's performance gets increasingly harder as the athlete approaches his ultimate." This results in a progressive scoring table that must be monitored to control the excess near the ultimate. This can be seen in power or exponential equations of today.
- The scores for different events should be comparable.
- There should be a scientific basis for any scoring system.[6]

These three interests have primarily motivated the development of the scoring tables since this time and as of today are still being examined. More recent advances in technology, such as pole vault poles, have unbalanced earlier versions of the tables, necessitating periodic review.

The Decathlon tables do provide some sort of adequate, scientific comparisons but only for those ten events. A separate set of tables known as the IAAF Scoring Tables was developed to apply to individual events as opposed to combined events. Last updated in February of 2005, these tables are the basis of the IAAF world rankings and include every major event for both men and women, indoor and outdoor, from the track to the road to field events to relays. These tables are simply a list of times for each event and the corresponding numerical score. These scores put elites in the 1,200 range and appear

[6] IAAF Scoring Tables for Combined Events, Apr. 2004, 20 May 2007 <http://www.iaaf.org/newsfiles/32097.pdf>.

to be similar to Purdy Points, but are not directly relatable. The bottom of the table, worth one point, is approximately 6 minutes and 50 seconds for a mile, making direct predictions and comparisons impossible for a significant group of athletes. The official tables can be found here: http://www.iaaf.org/downloads/scoringtables/index.html.

Another type of scoring tables is known as the WMA Tables as of 2006 (formerly the WAVA Tables). These tables are unique in that the focus in on performances with respect to athlete age, coining the term “age-grading.” All running events from 50 meters to 200 kilometers are included plus field events. These tables generally approach a different problem than my own. Instead of doing predictions or comparisons for different distances, these are primarily used for comparisons amongst different age individuals for the same event, often used to determine winners for road races. These tables first appeared in 1989 and went through a major revision and in 1994.[7] Since then, hundreds of age group records were set, prompting a major revision of the then WAVA tables into the current 2006 version. As of this writing, the 2006 WMA Age Graded Tables can be found here: http://www.masterstrack.com/news2006/agt2006.xls.[8] A web calculator utilizing the 2006 factors created by Howard Grubb can be found here: http://www.howardgrubb.co.uk/athletics/wmalookup06.html.

When a performance is age-graded, a factor based upon the input event and the age is multiplied by the original performance to calculate an “age-graded result.” This age-graded result represents an equivalent performance by an individual at the peak of his ability. If a 25 year-old runner used this table, a factor of 1 would likely be used and the

[7] Age grading running races, 28 Apr. 2006, 22 May 2007 <http://home.stny.rr.com/alanjones/AgeGrade.html>.

[8] Ken Stone, Age Graded Tables finally arrive! And we have 'em, 2006).

result would not be adjusted. In addition to the graded result, an “age-performance percent” is given as a percent of that performance to the corresponding age group’s record performance. This percent value can than be used to compare or predict performances across different events. However, this comparison/prediction method across events is not scientifically accurate; it has been previously discussed that the percentage of the performance has been shown to not exhibit this linear comparison. For example, a Division III collegiate athlete may run the 100 meter in 11 seconds, or 88.82% of the 2006 world record. Another athlete regarded of similar performance may run 15 minutes and 20 seconds for the 5000 meters, or 82.32% of the 2006 world record. The latter individual would have to run 14:12 to achieve 88.82%! This time would almost guarantee being a national champion. An interesting solution may be to utilize the age-graded result as an input into a more accurate model such as the one I propose in the Methodology section or the Purdy Points model.

2.2. Predictive Models

Predictive models are focused on the individual runner and how his or her performance changes will respect to the distance run. A different subset of predictive models exist but are often intended for a specific distance predictions, such as predicting a marathon race from a 5 kilometer race, are generally less known. These models receive no attention in my analysis but are briefly included for completeness, as my research is currently intended for track events and multiple distances. For future research these models may be used and/or interpolated for analysis.

2.2.1. Pete Riegel's Model

Pete Riegel, a research engineer and marathoner, published this model in Runner's World in August, 1977 under the title "Time Predicting."[9] His model, one of the most simplistic, has since been republished in *Runner's World* as recent as 1997 and 1999.

$$\text{Predicted Time} = \text{Original Time} \times \left(\frac{\text{New Distance}}{\text{Original Distance}} \right)^{1.06}$$

This formula roughly says that when the distance run doubles, the speed at which it is run will drop by about 4%. His model, however, suffers three main limitations as described by the "Time Predicting" article on Runner's World's UK website: it assumes appropriate training has been done for the distance, it assumes one does not have a significant bias towards speed or endurance, and that calculations become less accurate for times less than three and a half minutes and over four hours.[10] This indicates that the formula may not perform well for the mid-distance and sprinting track events, but should be effective for events above 3000 meters.

2.2.2. VO_2 Max Model

Perhaps the most interesting and physiologically based model is the VO_2 Max Model, often written as VO2 max or VO2max. When one exercises, one consumes oxygen to produce energy. As the level of effort increases, the oxygen consumption

[9] Explaining the Performance Predictors, 2007, 21 May 2007 <http://run-down.com/statistics/calcs_explained.php>.

[10] RW's Race Time Predictor, 2004, 22 May 2007 <http://www.runnersworld.co.uk/news/article.asp?UAN=1681>.

increases as well until the body maxes out its ability to deliver and utilize oxygen. This level is known as VO_2 max. Studies have shown that this is an important factor in distance running as oxygen consumption, by definition, is linked to aerobic exercise. Unfortunately this model does not predict well when for sprints and shorter distances as these performances are generally achieved though anaerobic means where oxygen consumption may not be a key indicator.

This value can be found by doing a number of tests including laboratory measurement, but it can also be estimated through calculations from a race performance. In 1979, Jack Daniels and Jimmy Gilbert published the book Oxygen Power: Performance Tables for Distance Runners which ultimately provided regression equations that relate oxygen consumption to velocity.[11] Jack Daniels, having trained many elite runners and having coached SUNY-Courtland to eight national team titles and 130 All-America awards, has been recently recognized as Runner's World "World's Best Coach" and "NCAA Cross Country Coach of the Century." [12] His formula for calculating VO_2 max from races with velocity in meters per second and time in minutes is as follows:

$$VO_2Max = \frac{-4.60 + 0.182 * velocity + 0.000104 * velocity^2}{0.8 + 0.189e^{-0.0128 * time} + 0.299 * e^{-0.193 * time}}$$

Once a VO_2 max value has been calculated, prediction is accomplished by solving backwards for time with a desired distance. As velocity equals distance divided by time, the only unknown left is time. Due to the complexity of the equations, the most common

[11] J. Daniels and J. Gilbert, Oxygen Power: Performance Tables for Distance Runners (1979).
[12] 'World's Best Coach' joins Center for High Altitude Training, 24 Mar. 2005, 22 May 2007 <http://www.hastc.nau.edu/events-pressrm-032405.asp>.

method of predicting in this manner is to utilize approximation methods for time until the predicted VO_2 max is within reasonable error of the calculated value.

2.2.3. Runpaces Model

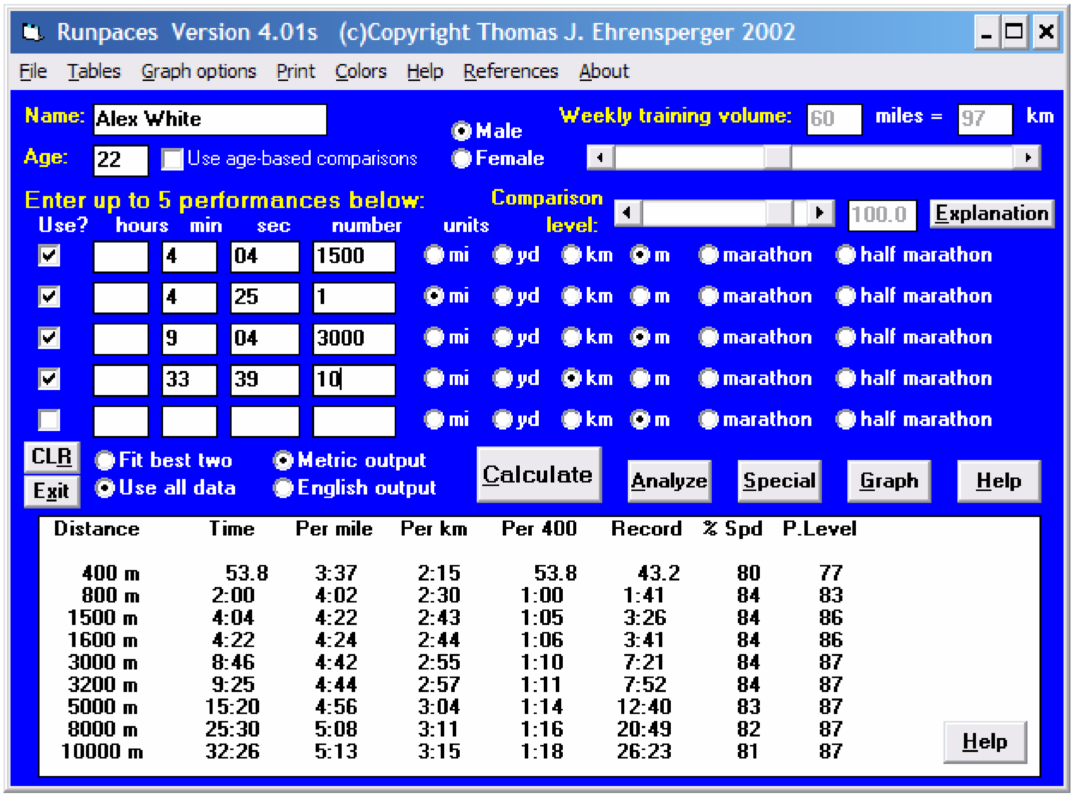
Runpaces is a lesser known model that operates within a graphical user interface as opposed to open mathematical equations or code. Developed by Thomas J. Ehrensperger, a runner and running enthusiast with a physics degree, it is currently at version 4.01, released in 2002.[13] His model is unique in that it utilizes multiple performances and incorporates age, gender, and weekly mileage to perform prediction. From my tinkering, his model performs reasonably over the gamut of distances from sprints to endurance races, but I am unable to validate it easily as I do not have access to the code or formulae. I have chosen to leave its analysis for future work but have mentioned it here for completeness and its unique nature.

Using his background he approached the problem from a physics and physiology standpoint from scratch using only existing models as reference. He began by separating performances into running into aerobic and anaerobic components modeled by power curves. He then estimated factors that affect the curve's shape based on physiological phenomena such as the accumulation of lactic acid in the blood, glycogen depletion, reaction time, and acceleration.[14]

In the following screenshot, I have entered my age, sex, training miles per week, and most importantly a set of four performances at different distances. The bottom of the

[13] Pace versus Distance Study, 21 Jun. 1997, 23 May 2007 <http://members.aol.com/eburger/study.html>.
[14] Runpaces 4.0: How it works, 28 Aug. 1999, 22 May 2007 <http://members.aol.com/eburger/#hiw>.

screen shows my predicted time for a set of standard distances from 400 to 10,000 meters. It is interesting to note that the model predicts times that are different for distances I have used to generate the results. This is attributed to the inclusion of multiple performances so that the resulting curve predicts these distances as well.


Runpaces Version 4.01s (c)Copyright Thomas J. Ehrensperger 2002
File Tables Graph options Print Colors Help References About
Explanation
CLR
Exit
Calculate
Analyze
Special
Graph
Help
Help

Name: Alex White

Age: 22 ☐ Use age-based comparisons

● Male ○ Female

Weekly training volume: 60 miles = 97 km

Comparison level: 100.0

Enter up to 5 performances below:

Use?	hours	min	sec	number	units
☑		4	04	1500	○ mi ○ yd ○ km ● m ○ marathon ○ half marathon
☑		4	25	1	● mi ○ yd ○ km ○ m ○ marathon ○ half marathon
☑		9	04	3000	○ mi ○ yd ○ km ● m ○ marathon ○ half marathon
☑		33	39	10	○ mi ○ yd ● km ○ m ○ marathon ○ half marathon
☐					○ mi ○ yd ○ km ● m ○ marathon ○ half marathon

○ Fit best two
● Use all data

● Metric output
○ English output

Distance	Time	Per mile	Per km	Per 400	Record	% Spd	P.Level
400 m	53.8	3:37	2:15	53.8	43.2	80	77
800 m	2:00	4:02	2:30	1:00	1:41	84	83
1500 m	4:04	4:22	2:43	1:05	3:26	84	86
1600 m	4:22	4:24	2:44	1:06	3:41	84	86
3000 m	8:46	4:42	2:55	1:10	7:21	84	87
3200 m	9:25	4:44	2:57	1:11	7:52	84	87
5000 m	15:20	4:56	3:04	1:14	12:40	83	87
8000 m	25:30	5:08	3:11	1:16	20:49	82	87
10000 m	32:26	5:13	3:15	1:18	26:23	81	87

Figure 4: Screenshot of Runpaces calculations

The following graph is produced by selecting the “Graph” button. The blue curve indicates my predicted running pace according to the model. The small circles indicate my input parameters for generation. The pink curve provides a predicted indication for where my “best” racing distances lie, seen at the apex here in the 5 to 8 kilometer range. The bright red curve indicates the pace required for the current world records. Notice the deep red curve, the percent of my predicted curve pace of the world record pace. It is clearly not linear, supporting the inadequacy of the WMA Age Graded Table method of prediction discussed earlier at the end of the Performance Tables subsection.

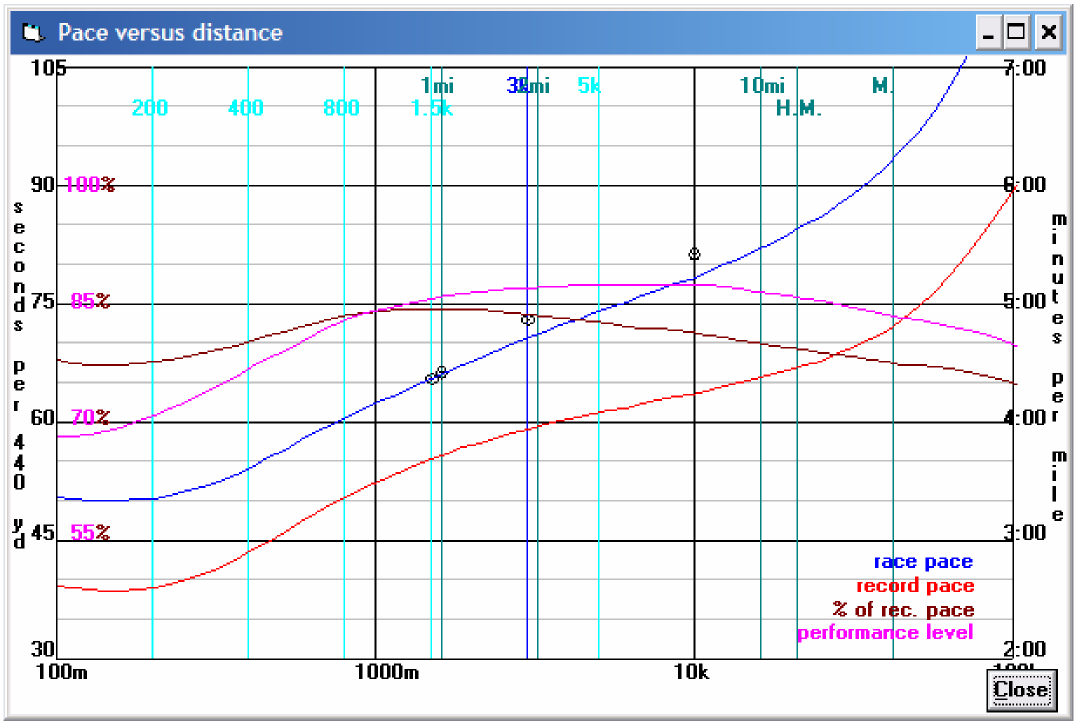


Figure 5: Screenshot of pace versus distance graph for Runpaces calculations

This data was gathered from the shareware version of the program. The author states that predictions down to 100 meters and additional analysis, among other features, are included in the full version.

2.2.4. Other Models

This subsection briefly describes other popular models that are not included in my analysis. One model is a set of formulae known as “Jeff Galloway's Magic Mile Race Prediction Formulas.” Jeff Galloway is a former US Olympian who has written a number of top-selling running books and is also a columnist for the popular *Runner’s World* magazine.[15] He claims that through working with over 170,000 runners he has compiled hundreds of performances and has “established a prediction formula based upon a one mile time trial.” From this time trial pace, paces for longer races are determined from the following calculations:

- Add 33 seconds for your pace for a 5K,
- Multiply by 1.15 for 10K pace,
- Multiply by 1.2 for half marathon pace,
- Multiply by 1.3 for marathon pace.[16]

There is one primary issue that has been recognized by such prediction methods. A one mile run can be more of an anaerobic race distance whereas long distance races are almost entirely aerobic. Generally, these races have a high dependency on three factors: running efficiency, VO2max, and lactate threshold.[17] These factors would not be adequately reflected in a one mile run.

Greg McMillan is a runner, exercise scientist, and coach who has developed his own predictive running calculator after finding that existing methods were not “specific enough.” During his college education, he completed senior and graduate theses regarding running performance and has actively studied the field of sport science.[18] His calculator produces a wide range of predictions from the 100 meter dash to the marathon and would be very suited for my analysis. However the formulae are not made public and therefore interfacing becomes difficult. Future analysis of this model through scripting

[15] Who is Jeff Galloway?, 2004, 21 May 2007 <http://jeffgalloway.com/about_jeff/index.html>.

[16] Jeff Galloway's Magic Mile Race Prediction Formulas, 2006, 21 May 2007 <http://jeffgalloway.com/resources/gallracepredict.html>.

[17] M. J. Joyner, "Modeling: optimal marathon performance on the basis of physiological factors," Journal of applied physiology (Bethesda, Md.: 1985) 70.2 (1991): 683-7.

[18] McMillan Running Coaching Staff, 2006, 21 May 2007 <http://www.mcmillanrunning.com/aboutus.htm>.

would be very interesting as from my experimenting it has performed well. In addition to providing predictions, the calculator offers suggestions for workouts and various training paces. This calculator can be found here: http://www.mcmillanrunning.com/rununiv/mcmillanrunningcalculator.htm.

2.3. Running Logs

The concept of logging running activities has been around for some time. The primary reason for their use is to track mileage. Most obviously, this provides motivation to run and continue running. Runners generally operate on a weekly basis (miles per week) for convenience. If one makes a large jump in mileage from one week to the next or runs too many miles, it greatly increases the chances for injury. By having this log, runners can track how many miles they have put on their shoes. There is a general rule that running shoes should only be worn for 500 miles as the cushioning properties diminish, also increasing the chances for injury[19].

All of this used to be recorded in spiral booklets. While getting the job done, it discourages analysis such as graphing running pace versus time or distance. With the rise of computing many applications have developed to provide additional features, such as analysis, that a pen and paper cannot. More recently, web-based running logs have grown in popularity as they are stored remotely and sharing becomes greatly simplified. However, the logs are succumbing to one of the downfalls of many computer applications – the battle between usability and usefulness. As the number of inputs increase, the less likely they are to be used.

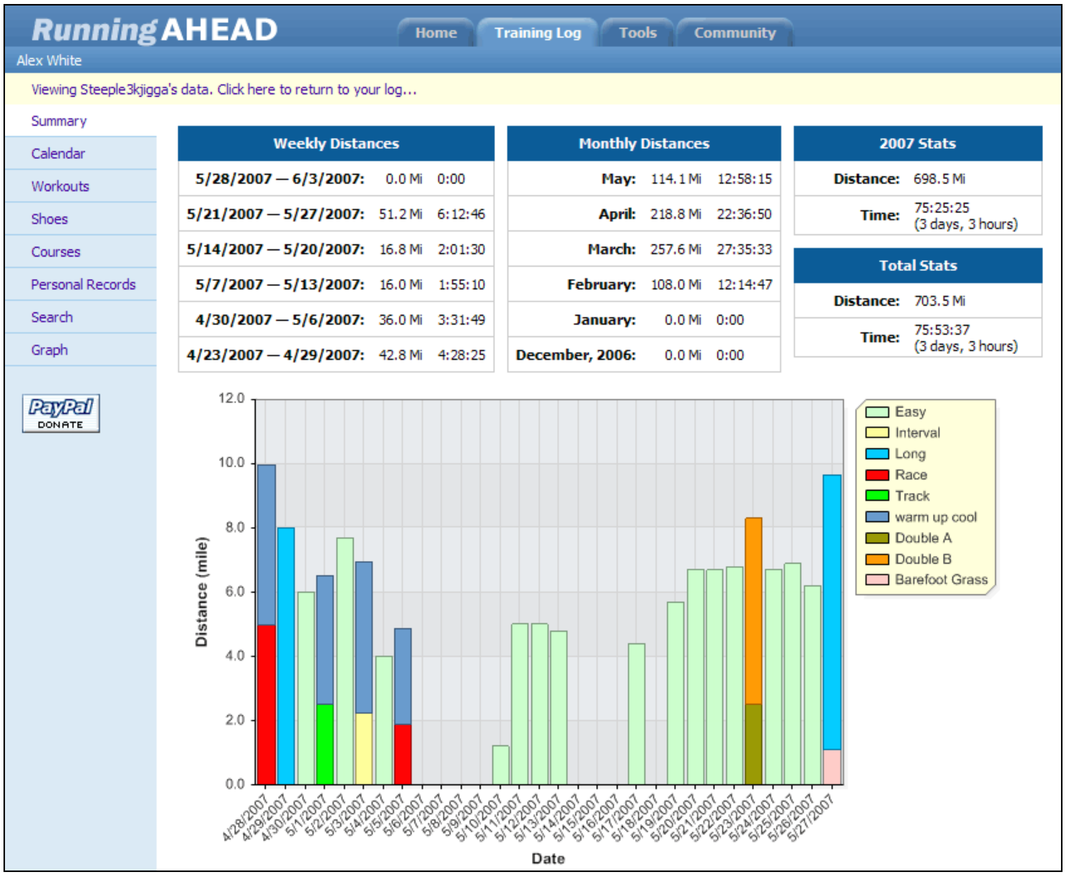
[19] Running: Preventing Overuse Injuries, Jul. 2005, 26 May 2007 <http://familydoctor.org/online/famdocen/home/healthy/physical/sports/147.html>.

Running logs are very optional in that there are few requirements for what needs to be entered to function properly, namely a date and a distance. This limits what is recorded to the amount of data a runner is willing to enter. Applications with poor usability ultimately have less data entered into them. As a result, data that may be useful to both analysts and users, such as sleeping habits or heart rates, may never get recorded. This is the primary issue that has motivated my development of a running log. Current running logs are not generally designed with the outside analyst in mind, either by making it cumbersome for users to enter data or by lacking input methods for possibly significant data. Additionally, many logs lack motivational features that would encourage users to continue use. By developing a log with the emphasis on usability and motivation for use, both the user and external analysts benefit. In this sense, the data useful on my behalf would come at no additional expense of the user. The following subsections describe a few popular running logs that I have had personal experience with. Their features and shortcomings are documented and have been used as a starting point for the development of my own log. This list is by no means complete as a large number of other running logs are available.

2.3.1. RunningAHEAD

RunningAHEAD is a relatively new free web-based running log started by Eric Yee, a graduate of Boston University, who wanted to make better use of the data provided by runners. The website for this log is http://www.runningahead.com and promotes itself though the motto: “Train. Analyze. Improve. Achieving goals through better information.” The log came online around 2005 and has been undergoing constant development since.

The homepage for a user provides a summary of miles run in recent weeks and months as well as a color-coded bar chart of recent runs. The user interface is overall very clean and intuitive with a simple and easy to follow color scheme. An example of a summary page is shown below.



Viewing Steeple3kjigga's data. Click here to return to your log...

Weekly Distances		
5/28/2007 — 6/3/2007:	0.0 Mi	0:00
5/21/2007 — 5/27/2007:	51.2 Mi	6:12:46
5/14/2007 — 5/20/2007:	16.8 Mi	2:01:30
5/7/2007 — 5/13/2007:	16.0 Mi	1:55:10
4/30/2007 — 5/6/2007:	36.0 Mi	3:31:49
4/23/2007 — 4/29/2007:	42.8 Mi	4:28:25

Monthly Distances		
May:	114.1 Mi	12:58:15
April:	218.8 Mi	22:36:50
March:	257.6 Mi	27:35:33
February:	108.0 Mi	12:14:47
January:	0.0 Mi	0:00
December, 2006:	0.0 Mi	0:00

2007 Stats	
Distance:	698.5 Mi
Time:	75:25:25 (3 days, 3 hours)

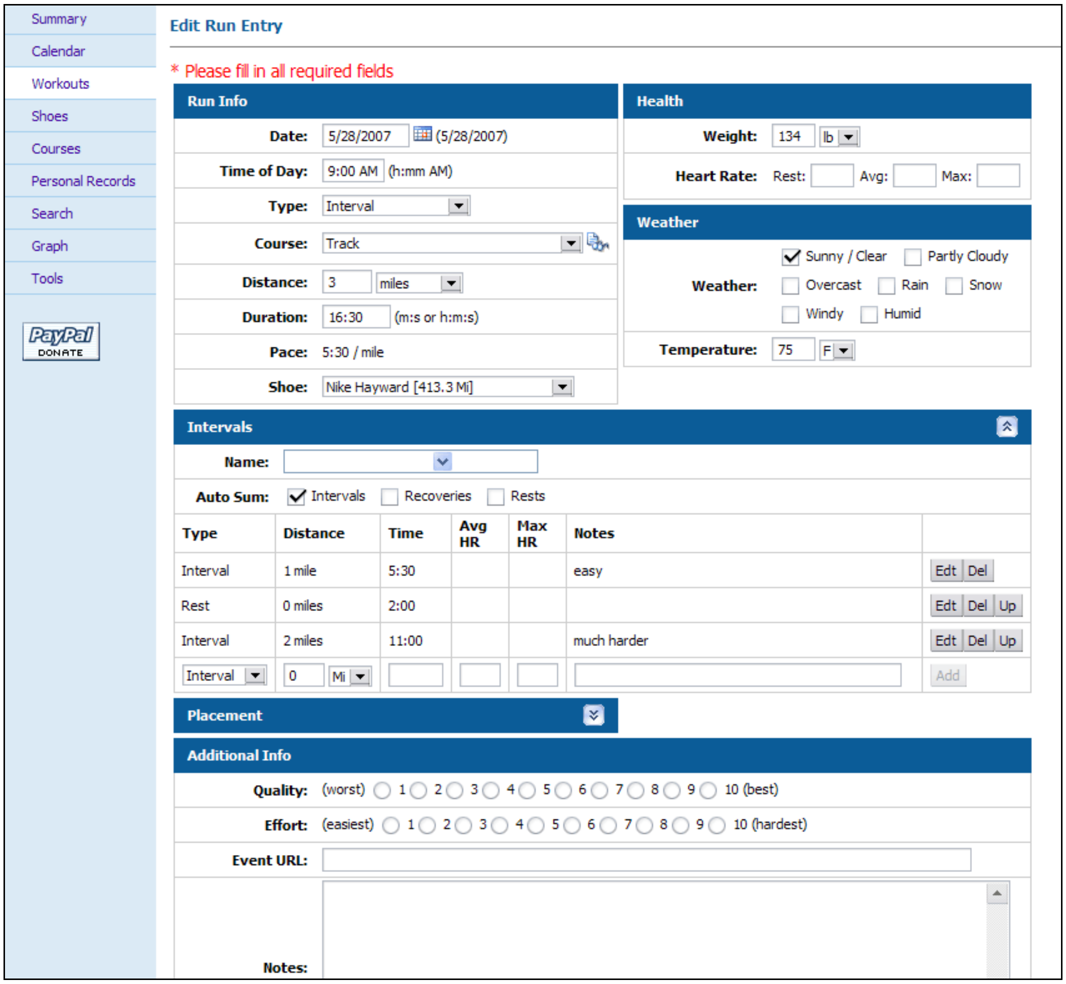
Total Stats	
Distance:	703.5 Mi
Time:	75:53:37 (3 days, 3 hours)

Figure 6: Summary page for the RunningAHEAD running log

This running log also takes advantage of community efforts to motivate runners and encourage log use. Users can create and join public or private groups where their logs can be shared along with any courses (also known as routes or runs) that may have created through the provided course creation tool. In addition to having a forum available to all users, each group has an individual forum as well.

The course creation tool utilizes the Google Maps API whereby users can click on the map to create a set of points defining the course. Their implementation includes addition features such as mile markers, out-and-back completion, and a more sophisticated “follow route back” for loops that share a common portion. In addition to accurate distance calculations, an elevation map can be viewed for the course.

To log a workout, a dialog is shown where a user can supply as much or little information as he or she chooses. An interesting feature is that repeat workouts can be input individually for a finer granularity of workout. An example workout input is shown below.



Edit Run Entry

* Please fill in all required fields

Run Info

Date: 5/28/2007 (5/28/2007)

Time of Day: 9:00 AM (h:mm AM)

Type: Interval

Course: Track

Distance: 3 miles

Duration: 16:30 (m:s or h:m:s)

Pace: 5:30 / mile

Shoe: Nike Hayward [413.3 Mi]

Health

Weight: 134 lb

Heart Rate: Rest: Avg: Max:

Weather

Weather: ☑ Sunny / Clear □ Partly Cloudy □ Overcast □ Rain □ Snow □ Windy □ Humid

Temperature: 75 F

Intervals

Name:

Auto Sum: ☑ Intervals □ Recoveries □ Rests

Type	Distance	Time	Avg HR	Max HR	Notes	
Interval	1 mile	5:30			easy	Edt Del
Rest	0 miles	2:00				Edt Del Up
Interval	2 miles	11:00			much harder	Edt Del Up
Interval	0 Mi					Add

Placement

Additional Info

Quality: (worst) ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 (best)

Effort: (easiest) ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 (hardest)

Event URL:

Notes:

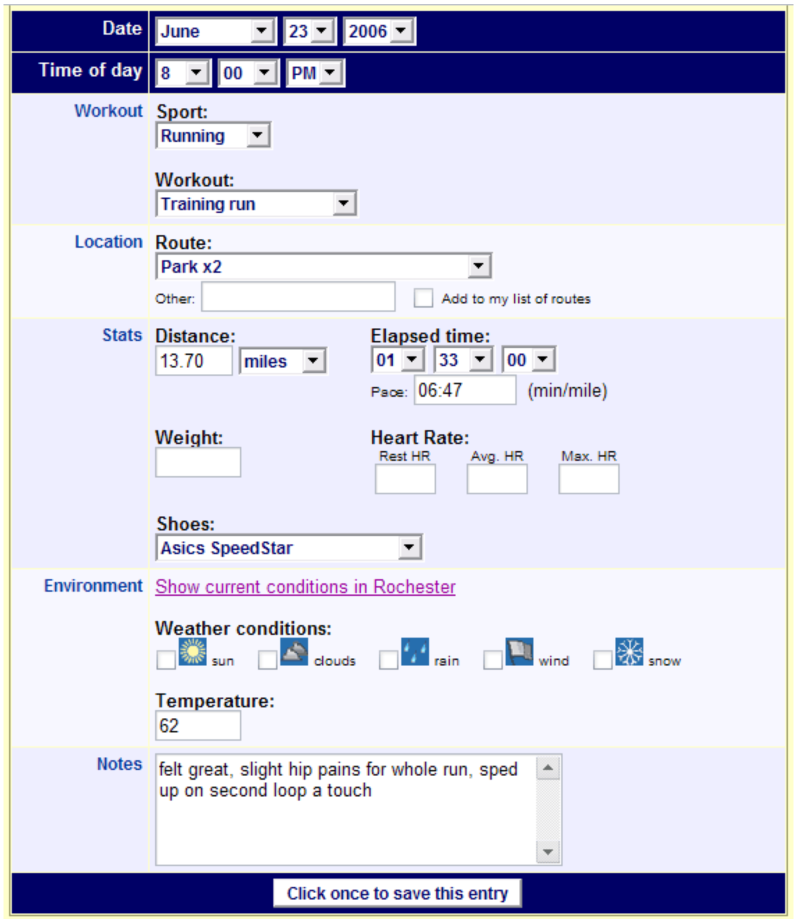
Figure 7: New workout entry for the RunningAHEAD running log

There are currently some issues with the input of new workouts. Once cannot input a pace and time to determine distance, or a pace and distance to determine time, which are often useful for users who estimate when they did not have a watch. For workouts where repeats are involved, inputting each takes more time than would be wanted. For example, while it fills some fields from previous inputs, it fails to recognize common patterns, such as interval/rest/interval/rest. It also does not recognize inputs that do not include a colon such as seconds.

An added feature is that while dedicated to running specifically, the log also natively supports other types of training such as swimming, cycling, and strength training. Users also have the option of defining their own types of training through the inputs are limited to basic fields such as distance and duration. Powerful but complicated graph and search features are provided for workouts. The user can create sets of shoes to input for each run, helping track the miles run on each pair. This log also provides interesting little features such as a cost per mile for shoes and automatically tracking of personal records. Health notes can be added as well that include information such as calories consumed and hours of sleep. This information can be useful for tracking but are time consuming to add in addition to a workout.

2.3.2. Cool Running

Cool Running, found at http://www.coolrunning.com, is a website that is dedicated to all things running, including training, races, results, and articles. It is self-described as “the complete online resource for runners of all ability” and “has been online since 1995, making it the longest-running commercial site dedicated to our sport."[20] It is known for its compilation of race results and its running log is also popular. The log is quite simple offering input to a basic set of features though a responsive user interface. Adding or editing workouts is quick, with weight, heart rate, shoes, and weather as the only additional options. However, having to input the time manually through drop-down boxes is cumbersome. Like RunningAHEAD, users cannot determine time or distance from pace. A screenshot of the input dialog is provided.



Date June 23 2006

Time of day 8 00 PM

Workout Sport: Running

Workout: Training run

Location Route: Park x2

Other: □ Add to my list of routes

Stats Distance: 13.70 miles

Elapsed time: 01 33 00

Pace: 06:47 (min/mile)

Weight:

Heart Rate: Rest HR Avg. HR Max. HR

Shoes: Asics SpeedStar

Environment Show current conditions in Rochester

Weather conditions: □ sun □ clouds □ rain □ wind □ snow

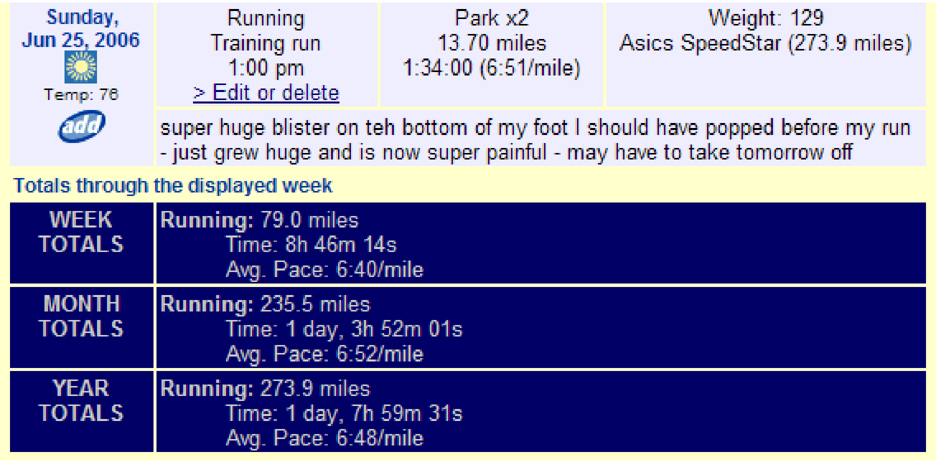
Temperature: 62

Notes felt great, slight hip pains for whole run, sped up on second loop a touch

Figure 8: New workout entry for the Cool Running running log

[20] About Cool Running, 2004, 26 May 2007 <http://www.coolrunning.com/engine/5/index.shtml>.

Running routes can be saved and used later but only store the distance. While other sports are included they are not natively supported as in RunningAHEAD, providing the same input set as the other activities. One of the better features of this log is that comments are displayed with each entry in the summary page. These comments have shown to be of special interest when others, such as coaches, examine the running logs. These logs can be viewed by an external link that does not require having an account. A screenshot of part of a summary page can be seen below.



Sunday, Jun 25, 2006 Temp: 76 add	Running Training run 1:00 pm > Edit or delete	Park x2 13.70 miles 1:34:00 (6:51/mile)	Weight: 129 Asics SpeedStar (273.9 miles)
	super huge blister on teh bottom of my foot I should have popped before my run - just grew huge and is now super painful - may have to take tomorrow off		
Totals through the displayed week			
WEEK TOTALS	**Running:** 79.0 miles Time: 8h 46m 14s Avg. Pace: 6:40/mile		
MONTH TOTALS	**Running:** 235.5 miles Time: 1 day, 3h 52m 01s Avg. Pace: 6:52/mile		
YEAR TOTALS	**Running:** 273.9 miles Time: 1 day, 7h 59m 31s Avg. Pace: 6:48/mile		

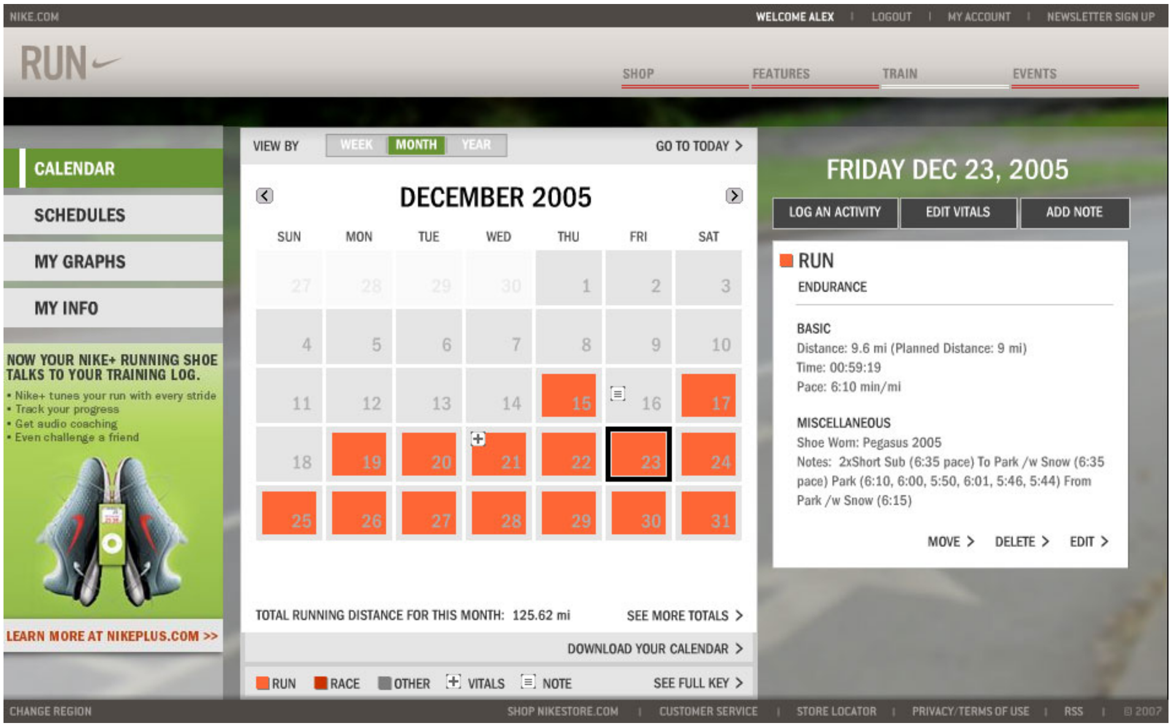
Figure 9: Summary for the Cool Running running log

Like RunningAHEAD, this log also finds personal records and tracks shoe mileage. Having the average pace available per week and month in the summary is interesting information to see how training is progressing. Only simple graphing features are available, such as a bar graph of miles per week, but users can export their logs as Excel spreadsheets for additional analysis.

2.3.3. Nike

The Nike running log is in a different class than the rest, being created by the shoe giant, but like the other logs it is free as well. This log, located at http://www.nike.com/nikerunning, has an included benefit of being able to sync with their Nike+ technology. Nike+ utilizes a sensor placed in the bottom of many of their shoe models to communicate with an adapter connected to an iPod Nano to track statistics such as time, distance, and calories burned.

The user interface is presented in a flash format. My usage has shown that this can consume a noticeable portion of system resources and cause slowdowns on some machines. Many commands take a noticeable amount of processing time with shown with an “updating data” icon. The summary page for this log, shown below, consists of a calendar view where each day is selectable, dynamically showing the details in the pane on the right.



DECEMBER 2005

SUN	MON	TUE	WED	THU	FRI	SAT
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

TOTAL RUNNING DISTANCE FOR THIS MONTH: 125.62 mi

🟥 RUN 🟥 RACE ■ OTHER ⊞ VITALS ▤ NOTE

FRIDAY DEC 23, 2005

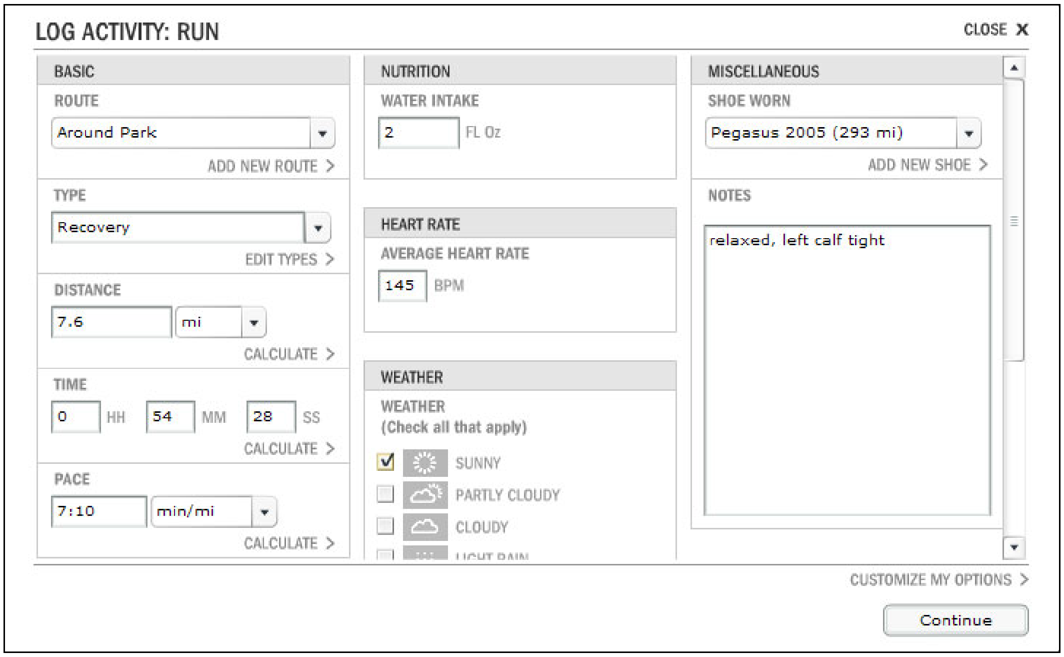
🟥 RUN
ENDURANCE

BASIC
Distance: 9.6 mi (Planned Distance: 9 mi)
Time: 00:59:19
Pace: 6:10 min/mi

MISCELLANEOUS
Shoe Worn: Pegasus 2005
Notes: 2xShort Sub (6:35 pace) To Park /w Snow (6:35 pace) Park (6:10, 6:00, 5:50, 6:01, 5:46, 5:44) From Park /w Snow (6:15)

Figure 10: Summary page for the Nike running log

Shoes can be added and modified, given a lifespan and users have the option of being alerted when a shoe reaches the end of this lifespan. Routes can be created similar to RunningAHEAD but users have the option of specifying many additional fields such as shade, scenery, and lighting, though it lacks a method to specify a visual map. While not being able to export the log information to Excel, one can export the calendar to Outlook, iCal, and Google Calendar. Like RunningAHEAD, the log natively supports many different activities and allows users to define their own. A big advantage of this log from the user's point of view is the ability to enable or disable a number of logging options, including nutrition, pre/post activities, and feeling. However, as most are disabled by default, the average user would not log this information and would therefore be unavailable to analysts.



LOG ACTIVITY: RUN

BASIC

ROUTE: Around Park

TYPE: Recovery

DISTANCE: 7.6 mi

TIME: 0 HH 54 MM 28 SS

PACE: 7:10 min/mi

NUTRITION

WATER INTAKE: 2 FL OZ

HEART RATE

AVERAGE HEART RATE: 145 BPM

WEATHER

WEATHER (Check all that apply)

☑ SUNNY
□ PARTLY CLOUDY
□ CLOUDY
□ LIGHT RAIN

MISCELLANEOUS

SHOE WORN: Pegasus 2005 (293 mi)

NOTES: relaxed, left calf tight

Figure 11: Entry page for the Nike running log

The form to add a run entry is shown in the preceding figure. Unlike the other logs, this one does allow the user to calculate the remaining field of distance, time, or pace given two of the three measures. The entry form is very similar to the other logs' entry forms but can be greatly supplemented through the "customize my options" button at the bottom. However, unlike RunningAHEAD, there is no native support for intervals.

3. Methodology

In the efforts to improve performance prediction for running, this project was divided into two distinct components. The first component is a process for creating a predictive model, which was manifested in thee different versions. In creating this process, a method for model validation was developed that could evaluate the performance of existing models as well as my own with real-world data. The second component is a running log application intended to build upon existing designs and, through its use, provide the developer or analyst with sufficient quantities of information to apply data mining techniques. The theory behind its development is that humans are incredibly complex and, possibly more importantly, unique. There are a number of training styles in use today, many of which were discovered in the past few decades. This indicates that we have yet to determine what may provide the best results, especially as different techniques may be more effectively for different individuals. By gathering a large enough pool of data through this running log application, it may be possible to examine these trends and validate training methods to maximize individual performance.

3.1. Predictive Models

The section draws upon information provided in the previous Background section. My original idea was to develop a performance model that would be more accurate than existing models over a larger spectrum of individuals. As many models are based upon the elite male performances, I felt that average runners and women were not adequately represented. This concept is lightly tackled by the IAAF scoring tables (incorporating women, though elite) and the WMA age-grading tables. In rethinking the concept, it became more interesting to study the problem of performance prediction as opposed to comparison. Performance prediction provides more useful benefits to the runner than a simple comparison does. Since most runners lie near the average group, and the elites have many models, I desired my predictive model to cater to those individuals.

Being unsure how to approach the prediction problem, I set it aside knowing that I would need to validate my resulting model in some manner. To accomplish this, I needed to find performances by specific individuals at various distances. These would be real-world points of data I could compare each model against. I came to the realization that I could use these individuals to predict performances, as well as validate. By finding runners who have performed similarly to my own at a given distance, it would be possible to use their performances in other events to produce predictions for myself.

The architecture diagram for this predictive model process is shown in the following figure. The process begins by crawling and parsing webpages with meet results into a database of individual results. These results are then processed by a model strategy and stored in a separate model database. This processing step is necessary as there are likely many data points for an individual at a given distance. This database is then queried by the presentation layer and returns the set of predictions. The original result database can then be utilized to validate the data-centric models produced by the different strategies as well as some of the other models discussed in the Background section.

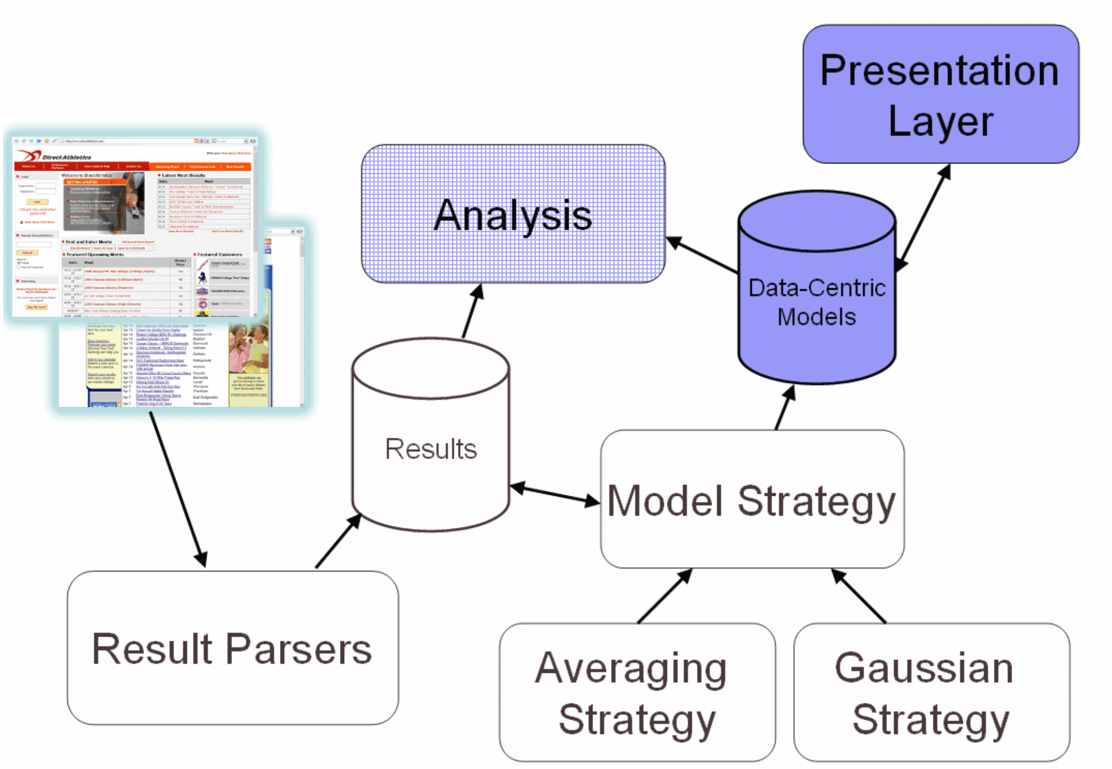


Figure 12: Architecture for predictive model development

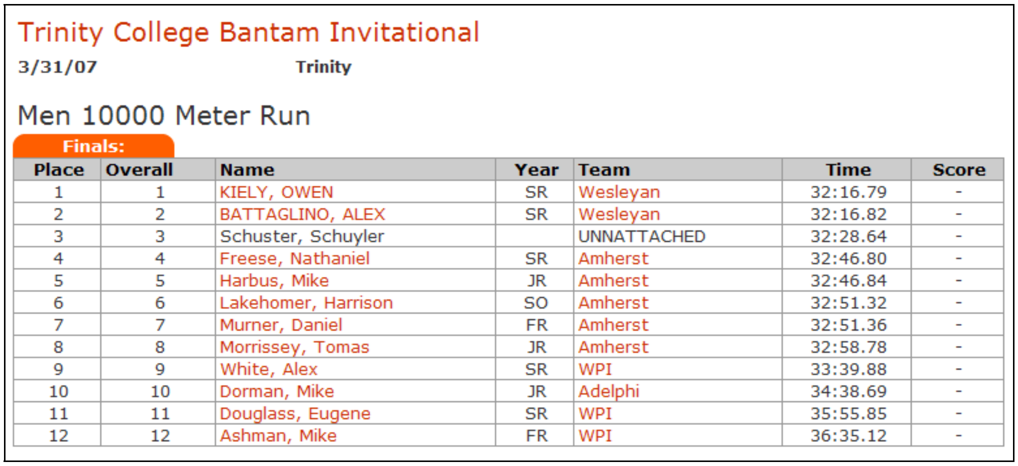
3.1.1. Result Parser

To create useful data-centric models, I needed to find the results of as many individuals as possible. There are a number of running result websites, one of which includes Cool Running, mentioned in the Background section. However another website, http://www.directathletics.com, was a perfect candidate for my search for performances. DirectAthletics is a Boston based company that provides online entry and meet result services to track and field teams, among other sports.[21] Their meet result system suited my needs perfectly as individuals have unique entities as the result of the online entry service.

[21] About Us, 2007, 29 May 2007 <http://www.directathletics.com/about_us.html>.

Unfortunately, retrieving individual performances was not as simple as pulling them out by person. DirectAthletics currently does not provide an adequate method of listing all teams or individuals. To work around this, I utilized their online meet result finder and crawled through every meet available for a specific state. Crawling though the meets proved very time consuming so the results for my analysis and model were limited to a subset of states. As collegiate athletes often run at meets in more than a single state (specifically those nearby), I included an adjacent area of states in my search.

To traverse these webpages, I wrote a dedicated result parser tailored to DirectAthletics that automatically collected result data. The parser was fed an array of state abbreviations that would serve as the entry points for meet listings. The entire process was completed though html connections. The parser then stepped into each meet and extracted performances for track and field events of note. Originally planning a performance model, I had included field events and hurdling events as well. The parser recorded the name, school, and sex of the athlete as well as the date, distance, and measure of the performance. A screenshot for a specific event below shows the formatted html.



Trinity College Bantam Invitational

3/31/07 Trinity

Men 10000 Meter Run

Finals:

Place	Overall	Name	Year	Team	Time	Score
1	1	KIELY, OWEN	SR	Wesleyan	32:16.79	-
2	2	BATTAGLINO, ALEX	SR	Wesleyan	32:16.82	-
3	3	Schuster, Schuyler		UNNATTACHED	32:28.64	-
4	4	Freese, Nathaniel	SR	Amherst	32:46.80	-
5	5	Harbus, Mike	JR	Amherst	32:46.84	-
6	6	Lakehomer, Harrison	SO	Amherst	32:51.32	-
7	7	Murner, Daniel	FR	Amherst	32:51.36	-
8	8	Morrissey, Tomas	JR	Amherst	32:58.78	-
9	9	White, Alex	SR	WPI	33:39.88	-
10	10	Dorman, Mike	JR	Adelphi	34:38.69	-
11	11	Douglass, Eugene	SR	WPI	35:55.85	-
12	12	Ashman, Mike	FR	WPI	36:35.12	-

Figure 13: HTML formatted event result from DirectAthletics

3.1.2. Result Database

To minimize network traffic and facilitate capture on my local Windows machine, the data were entered into an Access table. This table was later exported as a CSV file and imported into the WPI mySQL server for processing and analysis. A description of the storage table is provided below. As optimization was not a high priority for this proof of concept, all data fields were maintained. Additional fields were added for running speed and school type. Speed has been shown to behave more linearly with increases in distance reducing bias in comparisons. The school type field was set using commands similar to

```
UPDATE results SET schooltype = 'MS' WHERE school LIKE '% MS'
OR school LIKE '% MS %' OR school LIKE '% middle school' OR
school LIKE '% middle school %'
```

in efforts to simplify making finer grained analysis. The field events and other events not analyzed were removed from the table. To speed searching significantly, a key consisting of the `firstname`, `lastname`, and `school` fields was registered as they are the unique-per-individual fields.

3.1.3. Data-Centric Model Strategies

Once a database of results had been created, a process for retrieving results was developed. Simply querying this table for entries of the same distance where the speed was similar would produce biased results; there are a few problems with such a query. An individual may (and is likely to) run an event of the same distance a multiple of times. Depending on a number of factors, such as weather, sickness, or timer error, these performances may not accurately represent the performance of an individual. If a competitive athlete pulls a hamstring in the 100 meter dash but finishes, this query may incorrectly indicate that this runner may be similar to a recreational runner and distort the predictions. The ability of runners to improve over time is another issue. An individual running in the freshman year of high school may not have the strength or training that he or she has during the senior year. An example of an 800 meter runner demonstrating these issues, as extracted from my result database, is shown in the following figure.

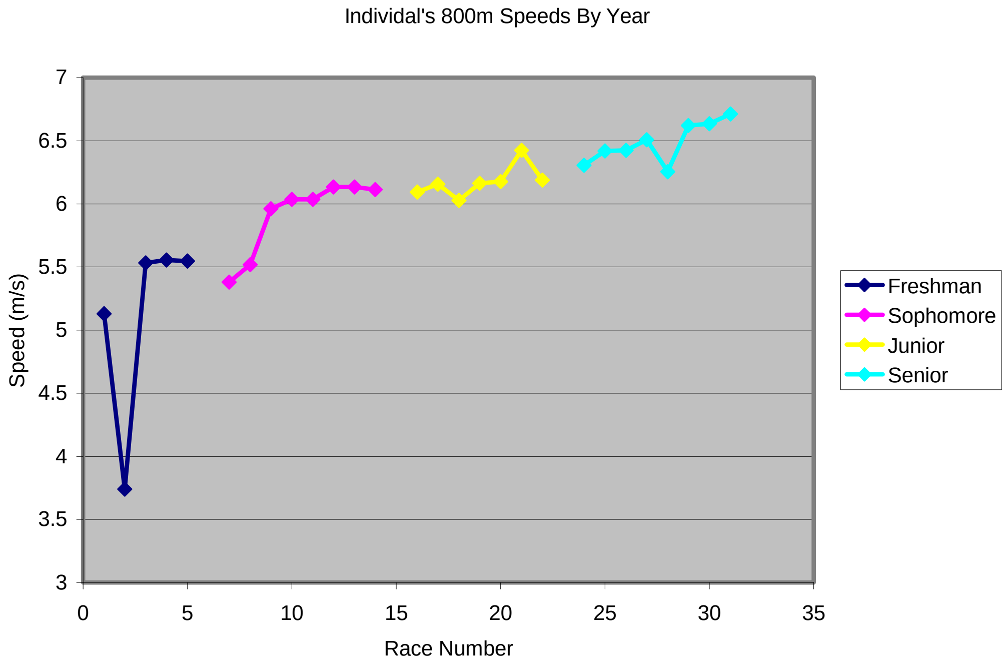


Figure 14: An individual athlete's performance over time

The athlete shown above has run the 800 meter dash on 31 occasions over his four years of competition. There are occasional dips in speed that are often more significant than jumps increases in speed. A clear trend of increasing speed is present.

My method for mitigating these problems consists of creating a new set of result tables whereby each individual will only have a single entry per distance. An individual is defined as a `firstname`, `lastname`, and `school` tuple. There are instances whereby a coach has entered an athlete's name differently, but these occurrences are more rare and separate from the scope of this project. The simplest method would be to average the set of results for each individual and distance combination. This strategy can be accomplished through a single SQL statement:

```
INSERT INTO average (lastname, firstname, school, sex, distance,
speed, std, num) SELECT lastname, firstname, school, sex2,
distance, AVG(speed), STDDEV(speed), COUNT(*) FROM results GROUP
BY lastname, firstname, school, distance
```

While simplistic, this strategy may be slightly unfair. It is very easy and thus more common for an athlete to perform worse than their "true" performance standard due to a number of reasons, some touched upon earlier. However, it is exceedingly more difficult to err on the side of faster as we are, unfortunately, limited by laws of physics and physiology. As a result, the slower performances have a greater impact on the average and may lead to less accurate predictions. By weighting performances according to some data distribution, it is possible to minimize the impact of these performances. One common distribution is the Gaussian or Normal curve which is often used to weight data points, such as in graphics rendering.

Though mySQL is very powerful, these distributions fall outside of the standard aggregate functions. To implement these strategies, I developed an abstract `CondensedGenerator` class that provided the necessary functions to retrieve individual performances. A `GaussianGenerator` class extends this, providing the underlying weighting of speeds. The Gaussian model requires two inputs to define the curve, a center point and standard deviation defining its spread. The actual calculations were done with the help of a Java class written by Robert Sedgewick and Kevin Wayne of Princeton University.[22] The center point is decided by the same average in the previous strategy. The standard deviation used is one half of that given by the mySQL aggregate function. This was subjectively chosen to narrow the band of the curve and further reduce the impact of outliers. The weighting pseudocode is described as follows:

```
for(individual’s performances at this distance)
     phi = Gaussian(perf_speed, avg_speed, std/2)
     denominator += phi
     numerator += perf_speed * phi
adusted_speed = numerator/denominator
```

Figure 15: Psuedocode for Gaussian weighting of results

Being implemented though an interface to mySQL this generation method is significantly slower. Other statistical distributions can be implemented through this method and are left for future work.

To resolve the other noteworthy issue of athletes changing performance over time, I utilized the date field to increase the resolution of the data set. As track and field seasons generally run from late December or January into the summer, the date field was simplified to the year value. With this method, each season in which an athlete competed is treated as a separate individual in attempt to further increase accuracy. This method, referred to as the “by Year” strategy, is independent of the previously discussed strategies and can be applied to any. For my analysis, discussed later, I have focused this method on

[22] Robert Sedgewick and Kevin Wayne, Introduction to Programming in Java: An Interdisciplinary Approach Addison Wesley, 2007), .

the averaging strategy. As before, this can be accomplished through a single SQL statement:

```
INSERT INTO averageByYear (lastname, firstname, school, sex,
year, distance, speed, std, num) SELECT lastname, firstname,
school, sex2, distance, year(date), AVG(speed), STDDEV(speed),
COUNT(*) FROM results GROUP BY lastname, firstname, school,
distance, year(date)
```

Predictions from these data-centric models are produced with an SQL query that accepts speed/distance pairs as inputs. My analysis was conducted on predictions from a single pair as this is how most existing performance/predictive models function. This is done by first finding individuals whose performance (defined by the strategies above) was similar to the input. To determine similarity, I choose to include all individuals within 0.05 meters/second of the input speed. For reference, this would equate to a delta of approximately 0.5 seconds in a 400 meter dash run as 60 seconds; 9 seconds in a 5000 meter run at 16 minutes. As a runner, I felt that these ranges are adequate variances to expect naturally. Being more precise may filter too many results creating unbiased weighting thereby decreasing accuracy and loosening them may also decrease accuracy. The following SQL query returns predictions for a runner at 2 minutes for an 800 meter dash and represents the interface to an outside presentation layer:

```
SELECT distance, avg(speed), std(speed), count(*) FROM average
WHERE (lastname, firstname, school) IN
   (SELECT lastname, firstname, school FROM average WHERE
   distance = 800 AND ABS(speed - 6.666666667) < 0.05 ORDER BY
   ABS(speed - 6.666666667) ASC)
GROUP BY distance;
```

Figure 16: Example SQL query to predict from a 2 minute 800 meter dash

3.1.4. Predictive Model Analysis

As previously mentioned, producing these data-centric models offers validation at little additional expense. To do this, I created a `ModelValidator` class that implemented the popular single-input performance/predictive models discussed in the Background section. To achieve the most accurate results, the validator uses the approach of dividing individuals by year as well. This translates to individuals predicting performances during the same season when they are at a similar fitness level. The process is described in the following pseudocode:

```
Randomize raw results
Get some raw results [optionally by some specification]
for ( each performance )
      Execute predictive queries but ignore that individual
            regardless of year
      Compare predicted results with performances at other
            distance by that individual (percent difference by
            time)
      Do same with existing models
      Write out data to files
            (input distance, predicted distance, predicted time,
            actual time, percent difference)
Repeat for each specification (women, high school, etc.)
```

Figure 17: Pseudocode for model analysis

3.2. Running Log

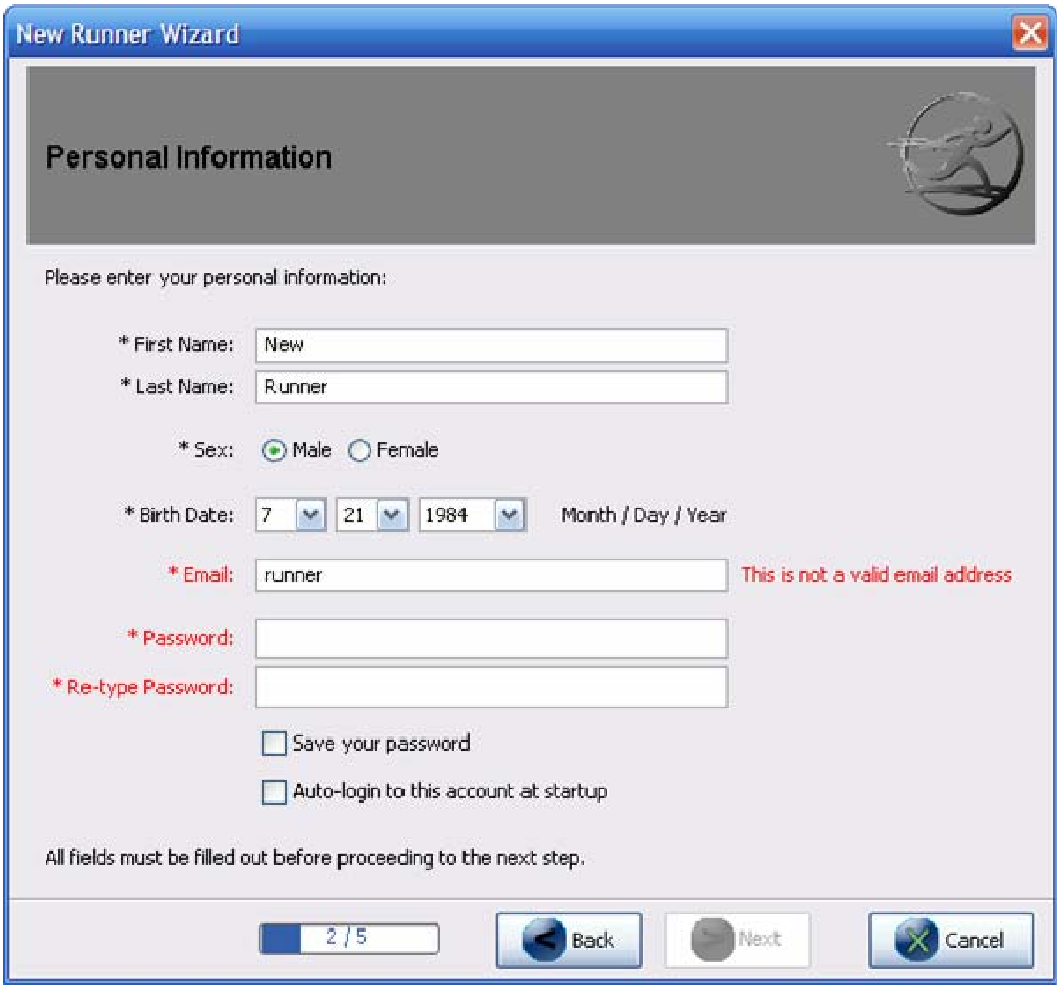
The running log component of this project was developed using previous designed as reference points. The two main goals of this design were to allow, but not force, input of a variety of pertinent data and to focus on usability. Having a user-friendly and usable interface encourages users to continue using the program. If this motivation did not exist, then the captured data would be less useful for analysis techniques. Ultimately, this involves auto-completion and other learned behavior mechanisms, many of which are beyond the scope of the project. As a starting point, attempts were made to produce a responsive, appealing running log that allows for simple storage and access for reusable items. I chose to develop this application in the Java programming language. This allows for a responsive interface by being more independent of an internet connection. Additionally, this permits offline usage and gives a standard application look-and-feel. It is also well supported by open source libraries for database interfacing and other features. To increase motivation for the application's use, I have included user groups and runner communication as a focus. Throughout the application, common UI design principles were followed such as consistency, informative feedback though status indicators, and dynamic error checking and simple error handling. The main application window uses tabbed panes to organize running log data; currently only the Routes pane is fully implemented.

3.2.1. New Runner Wizard

Upon first application startup, a wizard appears to help guide a new user to set up the application. This wizard was implemented though the use of a wizard library described in the article "Creating Wizard Dialogs with Java Swing" published on the Sun Developer Network by Robert Eckstein.[23] Its default functionality was extended to include a status bar informing the user about which panel is being viewed.

[23] Robert Eckstein, "Creating Wizard Dialogs with Java Swing," Sun Developer Network: Developer Technical Articles & Tips 2005.

I have implemented five wizard panels. The first consists of a splash image and introductory message. The second includes inputs for personal information as is seen in the screenshot below:



Personal Information

Please enter your personal information:

* First Name: New
* Last Name: Runner

* Sex: ● Male ○ Female

* Birth Date: 7 / 21 / 1984 Month / Day / Year

* Email: runner This is not a valid email address

* Password:
* Re-type Password:

□ Save your password

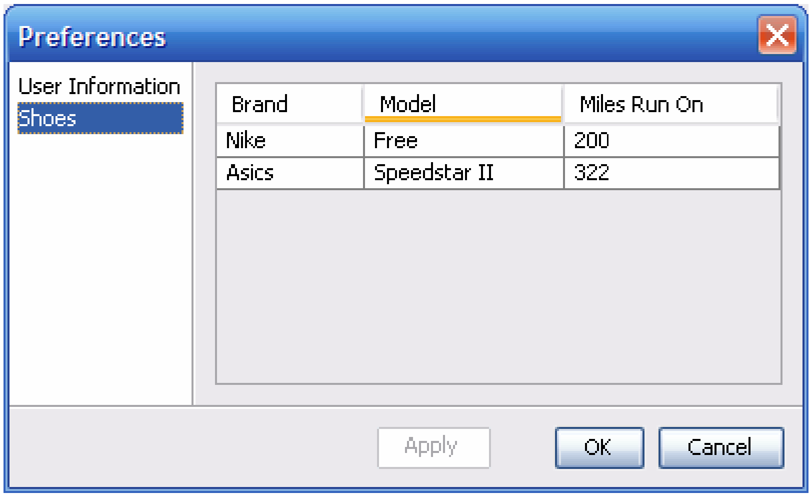
□ Auto-login to this account at startup

All fields must be filled out before proceeding to the next step.

Figure 18: New runner wizard

In the previous screenshot, we can see at the bottom that this is the second panel of five. The next button is currently disabled as not all of the required fields have been entered. The text boxes are attached to a listener that constantly monitors the fields for a valid response, following which the button is enabled. The “Email” field checks to see that a valid email address is entered – displaying an error message if this is not the case.

A similar rule applied to the password input. Each wizard panel shares the top title and bottom button components for consistency. The third panel requests inputs for statistics of a running nature such as weight, heart rate, weekly mileage, and running attitude. Tooltips are used to clarify data fields and options. To assist in measuring the resting heart rate, a graphical 20 second timer is placed in the panel. The fourth panel allows users to specify current pairs of shoes in an editable table. The final panel gives users an option of joining any number of existing running groups or creating their own. The groups and corresponding information are retrieved automatically from the mySQL database whose interface is provided by a singleton class. All persistent log data is stored in this database. Within the application, from the toolbar, the user may view and modify most of the information from a preferences dialog shown below.

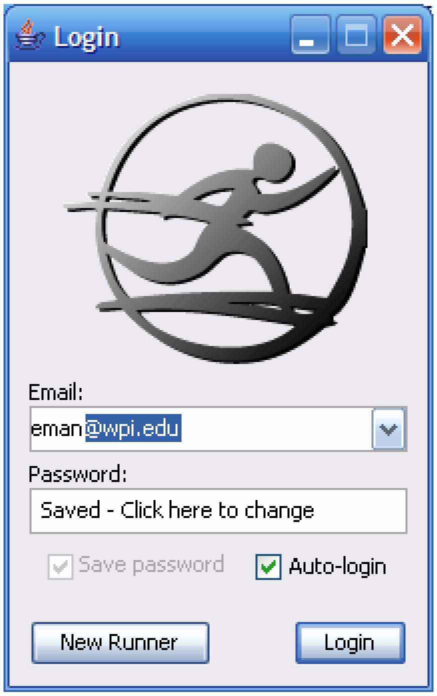


Brand	Model	Miles Run On
Nike	Free	200
Asics	Speedstar II	322

Figure 19: Preferences dialog

3.2.2. Login

If any user has already created an account via the wizard, the application begins with a login dialog modeled off of that provided by AOL Instant Messenger. This simplifies the management for multiple users as well as giving many users a familiar interface. The dialog, shown here, can be easily skipped by using the auto-login checkbox. Auto-completion, shown by the highlighted text, is performed on the editable dropdown box for any saved email address/user account. Useful persistent data such as window placement and size and user account login information are stored in a local properties file.



Email: eman@wpi.edu

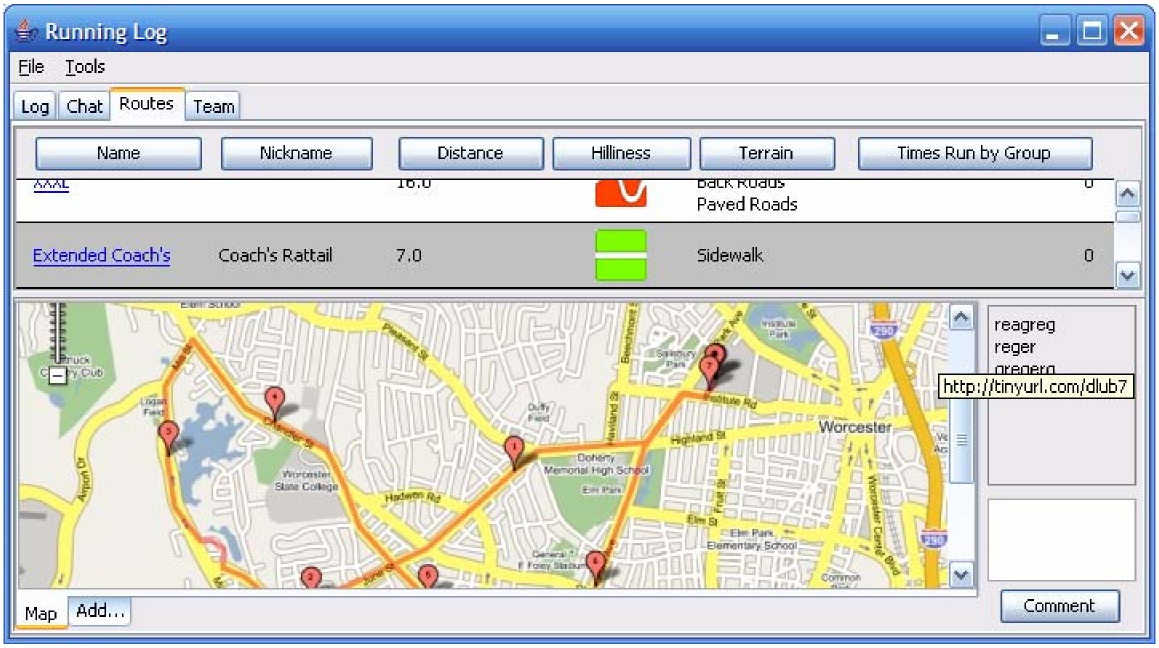
Password: Saved - Click here to change

☑ Save password ☑ Auto-login

Figure 20: Login dialog

3.2.3. Running Routes Panel

Most of my early development efforts went toward the Route panel once the new runner creation was finished. This panel displays all running routes available to that user. These may be either created by the user or shared by other members of any groups to which the user belongs. When a route is selected in the top panel, its images are retrieved along with any comments the route has from a mySQL table. Any images or comments added by the user are committed back to the table. A screenshot of the Routes panel is provided below.

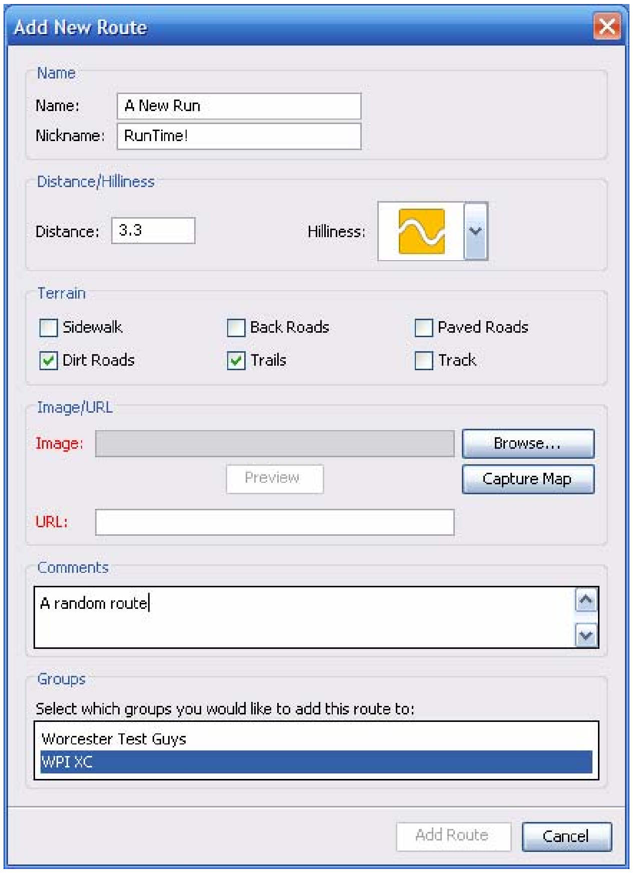


Name	Nickname	Distance	Hilliness	Terrain	Times Run by Group
XXXL		16.0		Back Roads Paved Roads	0
Extended Coach's	Coach's Rattail	7.0		Sidewalk	0

reagreg
reger
gregerg

http://tinyurl.com/dlub7

Figure 21: Screenshot of Routes panel



Name

Name: A New Run

Nickname: RunTime!

Distance/Hilliness

Distance: 3.3

Hilliness:

Terrain

☐ Sidewalk ☐ Back Roads ☐ Paved Roads

☑ Dirt Roads ☑ Trails ☐ Track

Image/URL

Image:

URL:

Comments

A random route

Groups

Select which groups you would like to add this route to:

Worcester Test Guys

WPI XC

Figure 22: Add route dialog

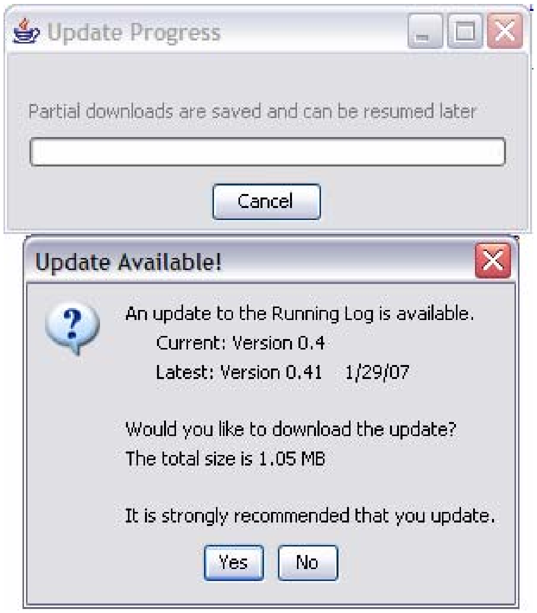
An add route dialog is implemented to allow users to describe and create routes. These can be shared with any group the user is a part of. The “Capture Map” button uses a handy Java robot that is designed to capture any type of Google Map, such as the Gmaps Pedometer, that exists in the window behind the application. When pressed, the application minimizes and the robot takes a screen capture.

Next, a processing engine discovers the borders of the map window, cropping and compressing the image. This implementation, while not as feature-rich as a dedicated Google Maps API, can utilize existing APIs effectively. The user also has the option of loading any local image though the image chooser dialog. These routes and images are stored in a mySQL table.

3.2.4. Auto-Updater

In preparing the running log for public release, I implemented an auto-updating mechanism for the running log. With an application in a beta or pre-beta stage, constant updates are needed for bug fixes and to add features. Many users may be unaware that updates are available or be turned-off by manual updates. This may ultimately discourage use, one of the issues to be avoided.

This operates by using a small bootloader application which handles download managements. Once new files are downloaded via http, it loads and executes the new running log jar file. To release a new version, the developer simply creates a new web folder with a version number greater than the previous and places the new files in that directory. The bootloader will discover the new version and determine which of its current files are out of date through size and last-modified comparisons, keeping the amount of data needing to be downloaded to a minimum.



Partial downloads are saved and can be resumed later

An update to the Running Log is available.
Current: Version 0.4
Latest: Version 0.41 1/29/07

Would you like to download the update?
The total size is 1.05 MB

It is strongly recommended that you update.

Figure 23: Auto-Updater dialog

4. Results and Analysis

This section describes the accomplishments made over the course of the year for both the predictive model and the running log application. Data-centric models were constructed using the discussed methods and validated with results data from the original result parser. I have described the functionality of the running log in the previous section and will focus on other metrics where applicable. It should be noted that the running log is under development and is by no means complete, though the previously documented features are near complete and are representative of its direction. With the concept so early in its lifecycle, I have not yet performed usability testing with outside parties. Additionally, testing was not a major concern as my primary focus was interface-based. I did however perform much exploratory testing and have polished the implemented components to a near-release status.

4.1. Predictive Model Generation

In traversing DirectAthletics for meet results, I included the entire New England region as well as New York, New Jersey, and Pennsylvania; DirectAthletics generally provided results from 2003 or so. Crawling over these states proved very time consuming and was a multi-overnight procedure. Traversing the HTML pages for a meet and parsing them consumed about two minutes on the average. This may be attributed to server-side bottlenecks associated with my constant access. In addition to being slow, the connection requests would occasionally hang. As a result, over 286,000 performances were recorded. However, approximately one-third of these performances were field events or hurdling events and were discarded leaving 186,687 running performances. The parsing code was 495 lines long, with approximately 50 lines per method.

Applying the model strategies to the raw performance data resulted in 88,675 unique individual/distance pairs offering an average of near two performances per distance, though over 1,000 individuals recorded at least 10 performances for the same distance. Applying the "by Year" strategy, where yearly performances are distinct, yielded approximately 20% more data points. Consequently, this decreased the average number of performances per distance to 1.5.

Additional parsing was done at a later date including Maryland, Delaware, West Virginia, Virginia, Kentucky, Ohio, Michigan, Indiana, and Illinois. However, increasing processing time due to database size and time constraints prevented model validation with this enhanced set of data. However, the model has been created and can be accessed. The expanded search resulted in almost 325,000 running performances, or 180,000 unique individual/distance pairs.

4.2. Predictive Model Validations

A `ModelValidator` object, implemented with approximately 300 lines of code, then applied the original results data to my models as well as those previously mentioned for a baseline comparison. Again, validation of my models rightfully ignored those individuals to be predicted when extracting and averaging data; it would be biased to use one's own performances to predict. Due to the long process of Gaussian model generation, a Gaussian model with the "by Year" strategy was not created.

In performing validation, I choose to analyze predictive performance for specific sets of individuals. These groups of runners include: males, females, middle school athletes, college athletes. Additionally, I examined predictions for distances "far way" from the original (defined as events more than two and a half times longer or shorter) and distances "closer" (defined as twice the distance or less – generally one event stepping). I limited the number of individuals to randomly select to 500 for analysis; predictions were done for all other performances by that individual, increasing the sample size as is noted in the following table. For example if my 800 meter dash performance was chosen and I've also run the 1500 meter twice and the 5000 meter once, predictions would be made for the three other races. The highlighted results are from my models.

Table 1: Model Validation (Male and Female)

	Male (1148 samples)		Female (1265 samples)	
Model	**Average % Error**	**Standard Deviation**	**Average % Error**	**Standard Deviation**
Average	3.14	2.92	3.57	4.53
Average By Year	3.05	2.76	3.63	4.52
Gaussian	3.15	2.91	3.55	4.48
Purdy	3.36	3.86	4.53	5.15
LS Purdy	3.90	4.03	5.63	6.32
Cameron	26.82	32.00	23.81	26.81
Riegel	7.32	6.24	6.80	6.29
VO_2 Max	6.60	7.08	7.25	7.07

For reference, a 3% error roughly translates into a prediction that was off by 1.5 seconds in a 400 meter dash to 30 seconds in a 5000 meter run. It is worth noting that all three models exhibit roughly the same behavior. This can be expected, especially for the very similar Gaussian and average models, as the same underlying data was used in their construction. The "By Year" strategy behaves similarly, but I gather that an enlarged data set would decrease its error and reduce standard deviation by achieving more resolution.

We can see that with track data for both males and females, the data-centric models predict more accurately and reliably than the leading Purdy Points model. For males the improvement is significantly less than with female runner. This can be attributed to Purdy's "neglect" of this factor. It is interesting to see that my models do not predict with the same average error for females as males, with significantly higher standard deviations. This may be a direct result of the larger variances in times observed by women athletes.

Dave Cameron's model fails horribly in this context. This is not necessarily a fault of the model as predictions under 400 meters (approximately half of all recorded events) are not directly applicable to his model. The VO_2 Max model does not appear to suffer from this as much, though still intended for longer distances. Finally, it is interesting to note that the more recent least squares Purdy Points model based on the velocity running curve does not perform as well as its older sibling. This is a trend that continues in the following results.

Table 2: Model Validation (Middle School and High School)

	Middle School (425 samples)		High School (953 samples)	
Model	**Average % Error**	**Standard Deviation**	**Average % Error**	**Standard Deviation**
Average	3.57	4.02	3.21	3.49
Average By Year	3.55	3.79	3.27	3.60
Gaussian	3.54	4.00	3.24	3.49
Purdy	4.28	4.17	3.64	3.65
LS Purdy	4.74	4.73	4.06	4.06
Cameron	16.52	12.56	23.31	27.02
Riegel	7.69	7.20	6.69	5.68
VO_2 Max	8.74	7.87	6.15	5.90

Again, the data show that my models outpace the popular Purdy Points. Note that my data-centric modeling process does offer more substantial improvements over existing models for "average" athletes such as those in the middle school range.

Table 3: Model Validation ("Far Away" and "Closer")

	"Far Away" (323 samples)		"Closer" (812 samples)	
Model	**Average % Error**	**Standard Deviation**	**Average % Error**	**Standard Deviation**
Average	3.18	2.73	3.20	3.23
Average By Year	3.28	2.91	3.20	3.18
Gaussian	3.18	2.67	3.23	3.23
Purdy	4.34	9.27	3.40	3.37
LS Purdy	5.58	10.13	3.93	3.70
Cameron	60.97	34.25	13.36	13.26
Riegel	9.55	7.71	5.84	4.67
VO_2 Max	7.89	8.86	5.72	4.78

The validation results for this type of test exhibit the flexibility of my method by predicting distances further away as precisely as those that are closer, a feat which none the other models analyzed here can claim as both the average errors and standard deviations fall off sharply. While I did limit the sample number to 500, the "Far Away" test produced fewer than this number of samples as some individuals did not have any performances that would classify in this range. A real-world example of predictions by these models is provided in Appendix C, using my own performance as input. While validation has shown excellent results for my data-centric model strategy, there are some additional issues that are discussed in the Future Work and Conclusions section.

4.3. Running Log

Originally, I had intended on producing a running log application complete enough to distribute to the public for use. However I was not able to dedicate enough development time to make this a reality in the limited time, especially as the important performance prediction aspect was the primary goal of this project. To add motivation for the application's use, I had intended on implementing a simple chat/message board interface where group members can communicate. The current running log showcases the HCI design considerations and my development for usability. In conducting exploratory testing, I had reported approximately 20 defects in Sourceforge of which most pertained to improper handling of unexpected user inputs. The majority of these defects, now fixed or mitigated, were related to the components which are now integrated in the application. I had written one set of JUnit tests for a number text field, which passed, that was used throughout the application.

To maximize reuse I constructed a library, mostly composed of swing UI objects such as panels. Additionally I created a package of data structures to facilitate communication between the application and the mySQL database. I began to implement a local hSQL database that would sync with the master mySQL data, allowing up-to-date offline usage. This feature was reduced in priority in favor of a functional workout entry system. In terms of metrics, the Auto-Updater and booloader portions were implemented in 648 lines of code over five classes. Averaged, 6.8 methods were implemented per class with 13.6 lines each. The main application is written with 7,321 lines of compilation code spread out in 78 classes over five packages: runninglog, runninglog.ui, runninglog.ui.wizard, runninglog.library, and runninglog.datastructures. The majority of code, 2,635 lines, is in the runninglog.ui package over 13 classes. Averaged overall, there are 6.6 methods per class with a length of 9.2 lines. The metrics plug-in for Eclipse was used to collect the metric data.

5. Future Work and Conclusions

As part of this project, I have demonstrated a method for producing a data-centric predictive model for track athletes. This technique could be applied to other running result sites such as Cool Running to not only increase the data size but include road races and other long distance races that are beyond the scope of the track. My research and analysis focused on simple averaging and a Gaussian weighting method to compress the data into the final predictive model. While my preliminary validation has shown only slight differences between the strategies, it may be possible for a different distribution to consistently outperform the test. Other asymmetrical distributions are likely more suited to the running data such as a gamma or chi distribution. Utilizing the SQL max() aggregate to select the fastest time and ignore the rest may be a possible alternative.

Characterizing a runner by a single performance is not revealing to the type or style of that runner. By utilizing additional queries, one could modify the data-centric models to accept multiple performances to give better predictions. For example, if I supply an 800 meter performance and a 5000 meter performance, it may be possible to match myself by both to a specific runner. To supplement this strategy, one could average matches for each performance separately with a lower weighting factor.

As the data that go into these models come from the real world, the majority of the results will be in the average range. This may cause the model to perform improperly for elite athletes as so few data points would be available. One could dynamically substitute a more suitable model (such as Purdy for elite athletes) for prediction. Another primary concern is that since this model is implemented though discrete values predictions for intermediary distances are unlikely. For example, the one mile run (approximately 1609 meters) is a very close race to the 1600 meter run. The current implementation treats these distances separately even though they could essentially be combined. Interpolating performances would not restrict prediction inputs and would serve to broaden to scope of the model.

Eventually, races supplied by running log users could be automatically integrated into the predictive models. Given enough data input into the running log application, data mining techniques could be put to use to further refine a predictive model that may incorporate a number of inputs beyond a single performance. As both a modeling and validation tool, I have had discussions with Thomas Ehrensperger about his Runpaces predictive model and Eric Yee of RunningAHEAD. My running log and model components may be of use for these currently published applications. The running log development has had a solid start and could be integrated with an exiting web-based running log such as RunningAHEAD as a standalone component.

For runners and fans, playing around with web-based performance calculators satisfies a craving for comparison, backed by data and science. The models serve multiple purposes: to compare individuals, to predict performances, and to reveal which performance may be the best. Current models are aging in a time where records continually fall, yet are still based on elite performances. Many fail to distinguish between different types of runners, such as by sex or age. An adaptive model such as the one proposed in this report attempts to cater to a variety of runners, such as the neglected average runner. Running is becoming a very popular sport and these models give us something to play around with without actually taking a step.

Appendix A Java code for Purdy Points Model

[Modified from: Patrick Hoffman]

```
/* Calculate the fraction of time from track curves.
 * It slows down the time from the tables
 */
private static double FractionOnTurns(double distance) {
      int laps, partLap, meters;
      double turnDistance;
      if (distance < 110)
            return 0;
      else {
            laps = Math.floor(distance/400);
            meters = distance - laps*400;
            if (meters <= 50)
                  partLap = 0;
            else if(meters <= 150)
                  partLap = meters - 50;
            else if (meters <= 250)
                  partLap = 100;
            else if (meters <= 350)
                  partLap = 100 + (meters - 250);
            else if (meters <= 400)
                  partLap = 200;
            turnDistance = laps*200 + partLap;
            return (turnDistance/distance);
      }
}

// Purdy Points function follows on next page
```

```
private static double PurdyPoints(double distance, double seconds) {
/*
 * Portuguese running table, distance, speed
 * Table was from World Record times up to 1936
 * They are arbitrarily given a Purdy point of 950
 */

      double portugueseTable[] = {
      40.0,11.000, 50.0,10.9960, 60.0,10.9830, 70.0,10.9620,
      80.0,10.934, 90.0,10.9000, 100.0,10.8600, 110.0,10.8150,
      120.0,10.765, 130.0,10.7110, 140.0,10.6540, 150.0,10.5940,
      160.0,10.531, 170.0,10.4650, 180.0,10.3960, 200.0,10.2500,
      220.0,10.096, 240.0,9.9350, 260.0,9.7710, 280.0,9.6100,
      300.0,9.455, 320.0,9.3070, 340.0,9.1660, 360.0,9.0320,
      380.0,8.905, 400.0,8.7850, 450.0,8.5130, 500.0,8.2790,
      550.0,8.083, 600.0,7.9210, 700.0,7.6690, 800.0,7.4960,
      900.0,7.32000, 1000.0,7.18933, 1200.0,6.98066, 1500.0,6.75319,
      2000.0,6.50015, 2500.0,6.33424, 3000.0,6.21913, 3500.0,6.13510,
      4000.0,6.07040, 4500.0,6.01822, 5000.0,5.97432, 6000.0,5.90181,
      7000.0,5.84156, 8000.0,5.78889, 9000.0,5.74211, 10000.0,5.70050,
      12000.0,5.62944, 15000.0,5.54300, 20000.0,5.43785,
      25000.0,5.35842, 30000.0,5.29298, 35000.0,5.23538,
      40000.0,5.18263, 50000.0,5.08615, 60000.0,4.99762,
      80000.0,4.83617, 100000.0,4.68988, -1.0,0.0 };
      double c1 = 0.20;
      double c2 = 0.08;
      double c3 = 0.0065;
      double v, d3, t3, d1, t1, t950, t;
      double a, b, k, d = 0.1;
      double points;
      int i;

      /* Get time from Portuguese Table */
      /* Find distance in table */
      for (i = 0; distance > d && d > 0; i += 2)
            d = portugueseTable[i];
      if (d < 1)
            return 0;                 /* Can’t find distance */
      i += -2;
      d3 = portugueseTable[i];        /* Get distance */
      t3 = d3/ portugueseTable[i+1];  /* Get time */
      d1 = portugueseTable[i-2];
      t1 = d1/portugueseTable[i-1];

      /* Use linear interpolation to get time of 950 pt. performance */
      t = t1 + (t3-t1)*(distance-d1)/(d3-d1);
      v = distance/t;
      /* Add the slow down from start and curves */
      t950 = t + c1 + c2*v + c3*FractionOnTurns(distance)*v*v;
      /* Calculate Purdy Points */
      k = 0.0654 - 0.00258*v;
      a = 85/k;
      b = 1 - 950/a;
      points = a*(t950/seconds - b);
      return points;
}
```

Appendix B Java code for Least Squares Purdy Points Model [Modified from: Patrick Hoffman]

```java
/*
 * Calculate least squares Purdy Points from 1970 world record
 * running curve.
 */
private static double LSPurdyPoints(double distance, double seconds) {
      double b1 = 11.15895;
      double b2 = 4.304605;
      double b3 = 0.5234627;
      double b4 = 4.031560;
      double b5 = 2.316157;
      double r1 = 3.796158e-2;
      double r2 = 1.646772e-3;
      double r3 = 4.107670e-4;
      double r4 = 7.068099e-6;
      double r5 = 5.220990e-9;
      double v, twsec;
      double a, b, k;
      double points;

      /* Calculate world record velocity from running curve */
      v = -b1 * Math.exp(-r1 * distance) + b2
                  * Math.exp(-r2 * distance) + b3
                  * Math.exp(-r3 * distance) + b4
                  * Math.exp(-r4 * distance) + b5
                  * Math.exp(-r5 * distance);

      /* Calculate world record time */
      twsec = distance / v;

      /* Calculate least squares Purdy Points */
      k = 0.0654 - 0.00258 * v;
      a = 85 / k;
      b = 1 - 1035 / a;
      points = a * (twsec / seconds - b);
      return points;
}
```

Appendix C My Personal Results

Table 4: My Personal Race Predictions

Distance	Actual	Average	Average By Year	Gaussian	Purdy	Riegel	VO2 Max
800m	2:00.2	2:00.57	2:01.55	2:00.6	1:57.97	2:05.95	2:04.63
1500m	4:05.24	4:05.35	4:05.36	4:05.37	4:05.24	4:05.24	4:05.22
1Mile	4:26.65	4:25.04	4:24.97	4:25.15	4:25.91	4:24.23	4:24.82
3000m	9:05.26	8:51.37	8:53.44	8:51.10	8:52.91	8:31.31	8:45.24
5000m	16:16.39	15:34.27	15:33.84	15:31.95	15:25.09	14:38.70	15:13.53
10000m	33:39.7	32:46.19	32:58.59	32:46.19	32:20.93	30:32.04	31:37.78

The table above shows the output of my data-centric models and other predictive models. My 1500 meter time of 4:05.24 was used as an input to the models, with the other distances as my desired distances for prediction. My models give more accurate predictions for most of the distances, though they and Purdy Points are very close for the one mile and 3000 meter runs. It is of note how far off my 5,000 meter and 10,000 meter predictions are for all models. This would be a clear indicator that my current optimal distance would lie in the 800 meter to 1 mile range. Though my models are still about 2% to 2.6% off, it is of note that the predictions are substantially better than those of the existing models.

References

Cameron, David F. "Time-equivalence Model: David F. Cameron Model." Jun 1998. <http://www.cs.uml.edu/~phoffman/cammod.html>.

Cool Sports. "About Cool Running." 2004. <http://www.coolrunning.com/engine/5/index.shtml>.

Daniels, J., and J. Gilbert. Oxygen Power: Performance Tables for Distance Runners. J. Daniels, J. Gilbert, (1979).

DirectAthletics. "About Us." 2007. <http://www.directathletics.com/about_us.html>.

Eckstein, Robert. "Creating Wizard Dialogs with Java Swing." Sun Developer Network: Developer Technical Articles & Tips 2005.

Ehrensperger, Thomas J. "Pace versus Distance Study." 21 Jun. 1997a. <http://members.aol.com/eburger/study.html>.

---. "Runpaces 4.0: How it works." 28 Aug. 1999b. <http://members.aol.com/eburger/#hiw>.

familydoctor.org Editorial Staff. "Running: Preventing Overuse Injuries." Jul. 2005. <http://familydoctor.org/online/famdocen/home/healthy/physical/sports/147.html>.

Galloway Productions. "Jeff Galloway's Magic Mile Race Prediction Formulas." 2006a. <http://jeffgalloway.com/resources/gallracepredict.html>.

---. "Who is Jeff Galloway?" 2004b. <http://jeffgalloway.com/about_jeff/index.html>.

Gardner, J. B., and J. G. Purdy. "Computer Generated Track Scoring Tables." Medicine and science in sports 2.3 (1970): 152-61.

Harlan, Natalie. "'World's Best Coach' joins Center for High Altitude Training." 24 Mar. 2005. <http://www.hastc.nau.edu/events-pressrm-032405.asp>.

Hoffman, Patrick. "Gardner-Purdy points." 2004. <http://www.cs.uml.edu/~phoffman/xcinfo3.html>.

IAAF Council. "IAAF Scoring Tables for Combined Events." Apr. 2004. <http://www.iaaf.org/newsfiles/32097.pdf>.

Jones, Alan L. "Age grading running races." 28 Apr. 2006. <http://home.stny.rr.com/alanjones/AgeGrade.html>.

Joyner, M. J. "Modeling: Optimal Marathon Performance on the Basis of Physiological Factors." Journal of applied physiology (Bethesda, Md.: 1985) 70.2 (1991): 683-7.

McMillan Running Company. "McMillan Running Coaching Staff." 2006. <http://www.mcmillanrunning.com/aboutus.htm>.

Run-Down.com. "Explaining the Performance Predictors." 2007a. <http://run-down.com/statistics/calcs_explained.php>.

---. "Performance Predictors." 2007b. <http://run-down.com/statistics/calc.php>.

Runner's World UK. "RW's Race Time Predictor." 2004. <http://www.runnersworld.co.uk/news/article.asp?UAN=1681>.

Sedgewick, Robert, and Kevin Wayne. Introduction to Programming in Java: An Interdisciplinary Approach. Addison Wesley, 2007.

Stone, Ken. Age Graded Tables Finally Arrive! and we have 'Em., 2006.

Track & Field News. "World Records - Men." 2 Dec. 2006. <http://www.trackandfieldnews.com/tfn/records/records.jsp?sex=M&typeId=0&listId=1>.